



MTS 4.0 Relational Gateway Database Data Dictionary



© 2010 by ACI Worldwide, Inc. All rights reserved.

All information contained in this documentation, as well as the software described in it, is confidential and proprietary to ACI Worldwide, Inc., or one of its subsidiaries, is subject to a license agreement, and may be used or copied only in accordance with the terms of such license. Except as permitted by such license, no part of this documentation may be reproduced, stored in a retrieval system, or transmitted in any form or by electronic, mechanical, recording, or any other means, without the prior written permission of ACI Worldwide, Inc., or one of its subsidiaries. ACI, ACI Worldwide, and the ACI product names used in this documentation are trademarks or registered trademarks of ACI Worldwide, Inc., or one of its subsidiaries.

Other companies' trademarks, service marks, or registered trademarks and service marks are trademarks, service marks, or registered trademarks and service marks of their respective companies.

Contents

Relationship File Tables	7
Account Tables	8
Account Table (ACCOUNT_T)	9
Account Address Table (ACCOUNT_ADDRESS_T)	16
Account Due-From Balance Table (ACCOUNT_DFM_BAL_T)	19
Account Drawdown Table (ACCOUNT_DRAWDOWN_T)	21
Account Subaccount Table (ACCOUNT_SUB_ACC_T)	23
Account Group Table (ACCOUNT_GRP_T)	26
Account Group Member Table (ACCOUNT_GRP_MBR_T)	30
Address Tables	32
Address Table (ADDRESS_T)	32
Address Destination Table (ADDRESS_DEST_T)	42
Address PIN Privilege Table (ADDRESS_PIN_PRIV_T)	49
Address Test History Table (ADDRESS_TST_HST_T)	50
Message Tables	53
Funds Transfer Message Tables	54
Message Table (MESSAGE_T).....	54
Message Credit Table (MESSAGE_CR_T)	73
Message Debit Table (MESSAGE_DR_T)	86
Message Accounting Table (MESSAGE_ACCTG_T).....	98
Message Destination Table (MESSAGE_DEST_T).....	104
Message History Table (MESSAGE_HIST_T)	111
Message Queue Table (MESSAGE_QUEUE_T)	113
Message Text Table (MESSAGE_TEXT_T)	115
Message Related Amount Table (MESSAGE_REL_AMT_T)	117
Message Returned Table (MESSAGE_RETURN_T)	118
Message Processing Rules Tables	120
Message Processing Rules Table (MESSAGE_PR_T)	120
Message Processing Rules Match Criteria Table (MESSAGE_PR_MCH_T).122	
Message Processing Rules Parameters Table (MESSAGE_PR_PRM_T)...123	
Processing Rules Parameter Values (MESSAGE_PR_PVL_T)	124
Repetitive Tables	125
Repetitive Table (REPETITIVE_T).....	125

Contents

Repetitive Credit Table (REPETITIVE_CR_T)	134
Repetitive Debit Table (REPETITIVE_DR_T).....	146
Auxiliary Tables	158
Auxiliary Address Table (AUX_ADDRESS_T)	159
Auxiliary ABA DDA Table (AUX_ABA_DDA_T).....	167
Auxiliary Notes Table (AUX_NOTES_T).....	168
Anticipation Template Tables	169
Anticipation Template Table (ANT_TMPL_T)	170
Anticipation Template Credit Table (ANT_TMPL_CR_T).....	180
Anticipation Template Debit Table (ANT_TMPL_DR_T).....	189
Bank Tables	198
Bank Table (BANK_T).....	199
Bank Clearinghouse Table (BANK_CLEARHOUSE_T)	203
Bank Special Table (BANK_SPECIAL_T).....	205
Bank Location Cost Table (BANK_LOC_COST_T)	208
Bank Fee Table (FEE_T).....	209
Change Tables	211
Change List Table (CHANGE_TBL_LIST_T)	212
Change Data Table (CHANGE_DATA_T)	213
Change Exclusion Table (CHANGE_TBL_EXC_T)	215
Channel Tables	216
Channel Table (CHANNEL_T)	217
Channel Address Type Table (CHANNEL_ADRTYP_T).....	220
Configuration Tables	221
Configuration Table (CONFIG_T)	222
Configuration Item Table (CONFIG_ITEM_T)	224
Configuration Item Description Table (CONFIG_ITEM_DESC_T)	226
Configuration Item Data Table (CONFIG_ITEM_DATA_T).....	228
Currency Table (CURRENCY_T).....	230
CUSIP Table (CUSIP_T)	232
Due From Monitor Tables	233
Due From Monitor Table (DFM_T)	234
Due From Monitor Balance Table (DFM_BAL_T)	236
Due From Monitor Derived Table (DFM_DERIVED_T).....	238
Due From Monitor Transactions Table (DFM_TRN_T).....	240
Exchange Rate Table (EXCHANGE_RATE_T)	242

Holiday Table (HOLIDAY_T)	245
Monitor Table (MONITOR_T)	247
Operator Tables	253
Operator Table (OPERATOR_T)	254
Operator Change Table (OPERATOR_CHANGE_T)	256
Operator Error Table (OPERATOR_ERROR_T)	259
Operator Log Table (OPERATOR_LOG_T)	261
Operator Logon Table (OPERATOR_LOGON_T)	263
Operator Privilege Table (OPERATOR_PRIV_T)	265
Operator Profile Amount Table (OPERATOR_PROFAMT_T)	267
Operator Template Table (OPERATOR_TMPL_T)	269
Period Date Table (PERIOD_DATE_T)	271
PIN Tables	272
PIN Table (PIN_T)	273
PIN Account Table (PIN_ACCOUNT_T)	277
PIN Function Override Table (PIN_FUNC_OVR_T)	279
PIN Function Table (PIN_FUNCTION_T)	281
Processing Rules Tables	283
Processing Rule Table (PRULE_T)	284
Processing Rules Match Table (PRULE_MATCH_T)	286
Processing Rules Parameter Table (PRULE_PARAM_T)	288
Processing Rules Parameter Values Table (PRULE_PARAM_VL_T)	289
Profile Table (PROFILE_T)	290
PSR Table (PSR_T)	292
Relational Gateway Link Table (RGWLINK_T)	301
Remote Tables	302
Remote Table (REMOTE_T)	303
Remote Parameter Table (REMOTE_PARAMETER_T)	308
Rules Tables	310
Rule Table (RULE_T)	311
Rule From Name Table (RULE_FROM_NAME_T)	314
Rule Nickname Table (RULE_NICKNAME_T)	316
Rule Pattern Table (RULE_PATTERN_T)	317
Rule Statistics Table (RULE_STATS_T)	318
SEPA Message Tables (MTS 4.0 and Higher)	319
Message ACH Regulatory Reporting Table (MESSAGE_ACH_RRPT_T)	320

Contents

Message ACH Table (MESSAGE_ACH_T)	321
Message Batch Control Messages Fund Table (MESSAGE_BCM_FUND_T)	
.....3 2 5	
Message Batch Control Messages Item Table (MESSAGE_BCM_ITEM_T)	
.....3 2 6	
Message BCM Table (MESSAGE_BCM_T)	327
Mandate Table (MANDATE_T)	329
Standing Order Tables	330
Standing Order Table (STANDING_ORDER_T)	331
Standing Order Log Table (STANDING_ORD_LOG_T).....	335
Static Operator Log Table (STATIC_OPR_LOG_T)	337
Stop Tables	338
Stop Table (STOP_T).....	339
Stop Name Table (STOP_NAME_T).....	342
Stop Token Table (STOP_TOKEN_T)	344
Supplement ID Table (SUPPLEMENT_ID_T).....	346
S.W.I.F.T. Authentication Table (SWIFT_AUTH_T).....	348
Template Tables	354
Template Table (TEMPLATE_T).....	355
Template Privilege Table (TEMPLATE_PRIV_T)	357

Relationship File Tables

The Relationship file (REL) tables in MTS store all information regarding your organization's customers. MTS uses this information for processing payments and administrative messages on behalf of customers.

The REL file enables your organization to build a single customer file for both domestic and international customers. These customers can include other banks, bank branches, internal bank departments, and corporate customers. MTS uses the information in this file during payment and administrative message processing. The REL file consists of an account file, an address file, and a group file. To build a customer file using REL, your organization needs to define account and address records for its customers. REL stores these records in an Account file and in an Address file.

Account Tables

The account tables contain information about how accounts handle messages.

Parent/child relationships

The Account table (ACCOUNT_T) is the parent table to the following child tables:

- Account Address (ACCOUNT_ADDRESS_T)
- Account Due-From Balance (ACCOUNT_DFM_BAL_T)
- Account Drawdown (ACCOUNT_DRAWDOWN_T)
- Account Subaccount (ACCOUNT_SUB_ACC_T)
- Account Group Member (ACCOUNT_GRP_MBR_T)

The Account Group table (ACCOUNT_GRP_T) is also a parent table to the Account Group Member table (ACCOUNT_GRP_MBR_T), which links the Account and Account Group tables.

Primary keys

The Account, Account Due-From Balance, Account Drawdown, and the Account Subaccount tables have the following primary keys in common:

- BANK_ID
- ID_TYPE
- ACCOUNT
- REL_TIME_STAMP

The Account Group and the Account Group Member tables have the following primary keys in common:

- GROUP_ID
- REL_TIME_STAMP

Account Table (ACCOUNT_T)

The Account table provides detailed information about accounts. Unlike other database maintenance tables, the Account table holds balance data and is therefore exported every time RGW runs.

Table Relationships

The following table lists the foreign keys in the Account table that relate to columns in the Account Address table:

These Account table columns...	Relate to these Account Address table columns...
BANK_ID ID_TYPE ACCOUNT	ACC_BANK ACC_TYPE ACC_KEY_ACC
CONCEN_BANK_ID CONCEN_ID_TYPE CONCEN_ACCOUNT	ACC_BANK ACC_TYPE ACC_KEY_ACC

Account Group table

The GROUP_ID column in the Account table relates to the GROUP_ID column in the Account Group table.

The BANK_ID and DAT_OWNER_BANK_ID columns in the Account table relate to the BANK_ID column in the Bank table.

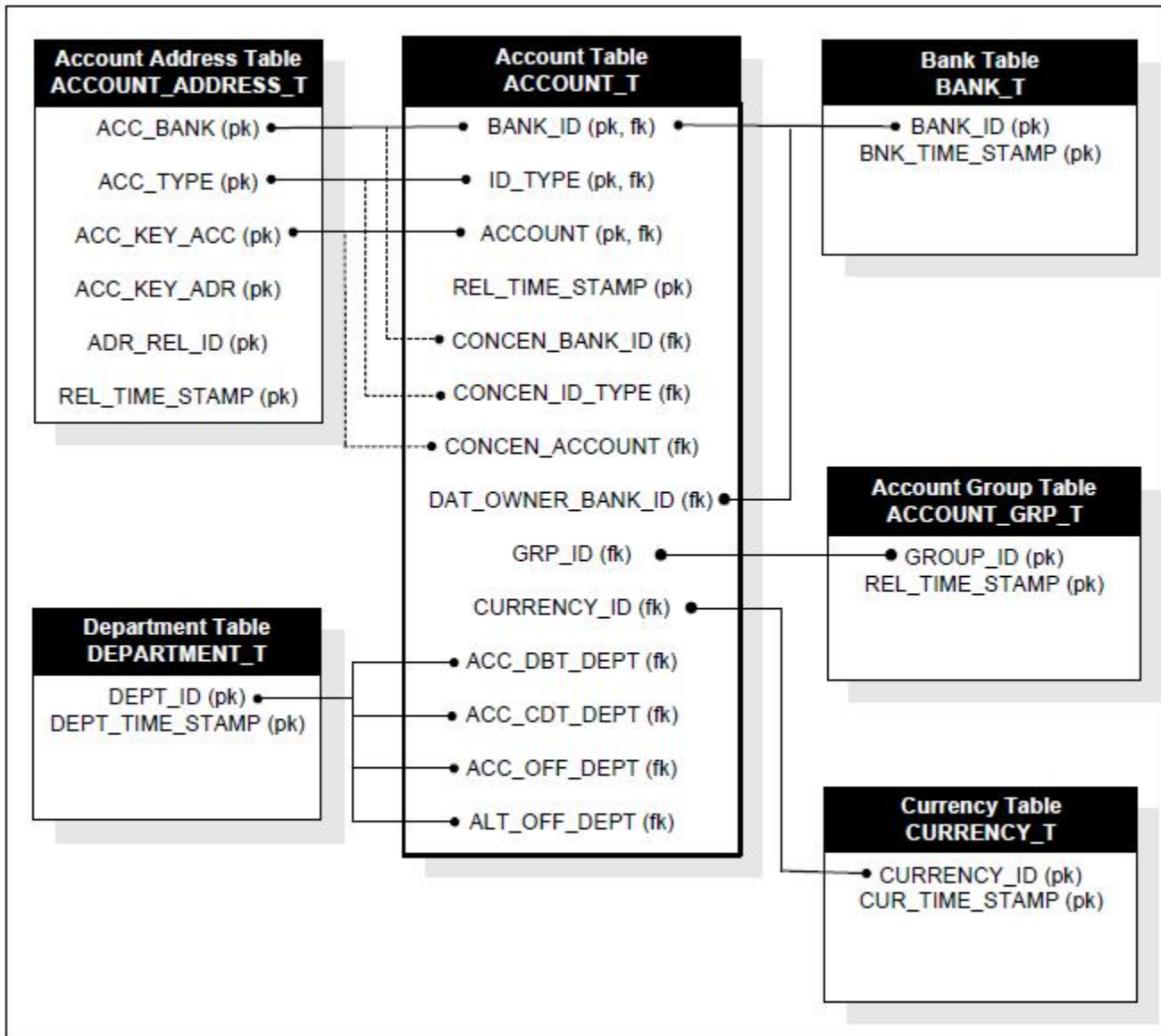
The CURRENCY_ID column in the Account table relates to the CURRENCY_ID column in the Currency table.

The following columns in the Account table relate to the DEPT_ID column in the Department table:

- ACC_DBT_DEPT
- ACC_CDT_DEPT
- ACC_OFF_DEPT
- ALT_OFF_DEPT

Relationship Diagram

This relationship diagram shows the Account table along with the tables related to it through foreign keys.



Column Descriptions

BANK_ID (pk, fk)	(varchar 3). Bank identifier of the bank that owns this account.
ID_TYPE (pk, fk)	(varchar 1). Account ID type; contains the following values: A (ABA)

	D (DDA) F (Foreign nostro account number) G (General ledger account number) P (CHIPS participant ID number) S (S.W.I.F.T.) V (Savings account number)
ACCOUNT (pk, fk)	(varchar 30). Account ID.
REL_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
DAT_FUNCTION	(vchar 7). Function that last accessed this account record; contains the following values: VFY ADD (Verify add) VFY DEL (Verify delete) VFY UPD (Verify update)
CREATOR	(varchar 10). Operator ID and bank ID of the operator who created the account record.
CREATE_DATE	(date). Date that the account record was created.
UPDATER	(varchar 10). Operator ID and bank ID of the operator who updated the account record.
UPDATE_DATE	(date). Date that the account record was updated.
VERIFIER	(varchar 10). Operator ID and bank ID of the operator who verified the account record.
VERIFY_DATE	(date). Date that the account record was verified.
DAT_OWNER_PROD_ID	(varchar 3). Product ID.
DAT_OWNER_BANK_ID (fk)	(varchar 3). Bank ID.
DAT_OWNER_LOC	(varchar 6). Location ID.
DAT_OWNER_CUST	(varchar 20). Customer ID.
LAST_USED	(date). Last date this account was used.
SPECIAL_IDBANK	(vchar 3). Bank-specific ID for the bank that owns the special account. A special account can be one of the following types of accounts: Fed G/L Fed Funds Fed Funds interest Multibank operations

Relationship File Tables

	Cross-bank accounting
SPECIAL_IDKEY	(varchar 20). Bank-specific name used to identify a special account.
CONCEN_BANK_ID (fk)	(varchar 3). Bank ID of the concentration account.
CONCEN_ID_TYPE (fk)	(varchar 1). Concentration account ID type; mapped from the Relationship File. This column contains the following values: D (DDA) F (Foreign nostro account number) G (General ledger account number) V (Savings account number)
CONCEN_ACCOUNT (fk)	(varchar 30). Concentration account ID.
GRP_ID (fk)	(varchar 10). Group ID if the account belongs to a group of accounts.
WE_INHIB_THEM	(varchar 1). CHIPS bilateral information.
THEY_INHIB_US	(varchar 1). CHIPS bilateral information.
ERROR_INHIB	(varchar 1). Indicates whether a CHIPS bilateral information error exists.
CHP_RCV_LIMIT	(number 21.3). Daily limit set by the clearing house on the amount received from another CHIPS participant.
CHP_SND_LIMIT	(number 21.3). Daily limit set by the clearing house on the amount sent to another CHIPS participant.
ACC_MINIMUM	(number 21.3). Minimum observed account balance at the start-of-day.
ACC_SOD_CHF	(number 21.3). Start-of-day clearinghouse funds balance.
ACC_SOD_LEDGER	(number 21.3). Start-of-day ledger balance.
ACC_SOD	(number 21.3). Start-of-day account balance.
GRP_MINIMUM	(number 21.3). Minimum observed group balance at the start-of-day.
GRP_SOD_CHF	(number 21.3). Start-of-day clearinghouse funds balance for a group.
GRP_SOD_LEDGER	(number 21.3). Start-of-day ledger balance for a group.
GRP_SOD	(number 21.3). Start-of-day balance for a group.
CURRENCY_ID (fk)	(varchar 3). Currency ID.
CONCEN_ACCT_FLAG	(varchar 1). Flag indicating whether this account is a concentration account (Y) or not (space or blank)

ACC_NAME	(varchar 35). Account name.
ADR1	(varchar 35). First line of the address.
ADR2	(varchar 35). Second line of the address.
ADR3	(varchar 35). Third line of the address.
ZIP	(varchar 10). Postal code.
COUNTRY_CODE	(varchar 2). Country code.
PHONE	(varchar 20). Phone number.
EXT	(varchar 4). Telephone extension.
SPEED	(varchar 4). Speed dial number.
ACC_DESC	(varchar 20). Account description.
ACC_CLASS	(varchar 6). Site-defined account class column used for site-specific applications; mapped from the CLASS field on the Relationship File (REL) Account File Screen 1.
ACC_CITY	(varchar 15). Site-defined account city; used for site-specific applications.
ACC_DBT_DEPT (fk)	(varchar 20). Site-defined account debit department column used for site-specific applications.
ACC_CDT_DEPT (fk)	(varchar 20). Site-defined account credit department column used for site-specific applications.
ACC_PROD_CODES	(varchar 20). Site-defined account product codes column used for site-specific applications.
CREDIT_RATING	(varchar 2). Site-defined account credit rating column used for site-specific applications.
CUSTOMER_CODE	(varchar 10). Customer code or other customer identification number.
PARENT_CODE	(varchar 10). Site-specific code used for site-specific purposes.
REVOCABLE_FLAG	(varchar 1). Flag indicating whether funds are revocable; contains the following values: Y (revocable) or blank (non-revocable).
SYSTEM_OF_RECORD	(varchar 3). System of record when the balance authorization is from another system; values are customer-defined.
CHARGES_DBT	(varchar 1). Flag indicating whether debit charges apply; relates to the CHG:DB field on the Funds Transfer (FTR) screen. Values are bank-defined in configuration files, determined from user-defined routines for fee charges.

Relationship File Tables

CHARGES_DBT_COMM	(varchar 1). Flag indicating whether debit charges apply for commission; relates to the CHG:COM field on the Funds Transfer (FTR) screen. Values are bank-defined in configuration files, determined from user-defined routines for fee charges.
CHARGES_DBT_CBL	(varchar 1). Flag indicating whether debit charges apply for cable; relates to the CHG:CAB field on the Funds Transfer (FTR) screen. Values are bank-defined in configuration files, determined from user-defined routines for fee charges.
CHARGES_CDT	(varchar 1). Flag indicating whether credit charges apply; relates to the CHG:CD field on the Funds Transfer (FTR) screen. Values are bank-defined in configuration files, determined from user-defined routines for fee charges.
CHARGES_CDT_COMM	(varchar 1). Flag indicating whether credit charges apply for commission; relates to the CHG:COM field on the Funds Transfer (FTR). screen. Values are bank-defined in configuration files, determined from user-defined routines for fee charges.
CHARGES_CDT_CBL	(varchar 1). Flag indicating whether credit charges apply for cable; relates to the CHG:CAB field on the Funds Transfer (FTR). screen. Values are bank-defined in configuration files, determined from user-defined routines for fee charges.
ACC_OFF_EMP_ID	(varchar 10). Employee ID of the bank officer responsible for the account.
ACC_OFF_PHONE	(varchar 20). Phone number of the bank officer responsible for the account.
ACC_OFF_DEPT (fk)	(varchar 20). Department of the bank officer responsible for the account.
ACC_OFF_NAME	(varchar 25). Name of the bank officer responsible for the account.
ALT_OFF_EMP_ID	(varchar 10). Employee ID of the alternate bank officer responsible for the account.
ALT_OFF_PHONE	(varchar 20). Phone number of the alternate bank officer responsible for the account.
ALT_OFF_DEPT (fk)	(varchar 20). Department of the alternate bank officer responsible for the account.
ALT_OFF_NAME	(varchar 25). Name of the alternate bank officer responsible for the account.
CREDIT_LIMIT	(number 21.3). Account credit limit.
DAYLIGHT_OD	(number 21.3). Account daylight overdraft limit.
DAYLIGHT_OD_EXP	(date). Account daylight overdraft expiration date.
PREADV_LIMIT	(number 21.3). Account preadvise limit.
PREADV_LIMIT_EXP	(date). Account preadvise limit expiration date.
TARGET_PERIOD	(varchar 1). Period for the target balance set by a bank on their customers' accounts. The following values can appear in this column: D (Day)

	W (Week) F (Fortnight or two weeks) M (Month) Y (Year)
TARGET_BALANCE	(number 21.3). Target account balance set by a bank on their customers' accounts; shown in account position reports.
TXN_LIMIT_CUR	(varchar 15). Maximum transaction amount for an account in a specific currency.
TXN_LIMIT	(number 21.3). Transaction dollar limit for the account.
NON_RPTV_LIMIT	(number 21.3). Non-repetitive callback limit.
RPTV_LIMIT	(number 21.3). Repetitive callback limit.
DLY_RLS_HRS	(varchar 2). The hour portion of the time until which MTS holds auto-hold payments.
DLY_RLS_MNS	(varchar 2). The minute portion of the time until which MTS holds auto-hold payments.
DBT_HOLD_FLAG	(varchar 1). Flag indicating whether an account's debit transactions will go to risk.
CDT_HOLD_FLAG	(varchar 1). Flag indicating whether an account's credit transactions will go to risk.
RISK_SPEC_INSTR1	(varchar 35). First line of risk special instructions.
RISK_SPEC_INSTR2	(varchar 35). Second line of risk special instructions.
DBT_SPEC_INSTR1	(varchar 35). First line of debit special instructions; displayed during payment entry when the account is used as a debit party.
DBT_SPEC_INSTR2	(varchar 35). Second line of debit special instructions; displayed during payment entry when the account is used as a debit party.
DBT_SPEC_INSTR3	(varchar 35). Third line of debit special instructions; displayed during payment entry when the account is used as a debit party.
CDT_SPEC_INSTR1	(varchar 35). First line of credit special instructions; displayed during payment entry when the account is used as a credit party.
CDT_SPEC_INSTR2	(varchar 35). Second line of credit special instructions; displayed during payment entry when the account is used as a credit party.
CDT_SPEC_INSTR3	(varchar 35). Third line of credit special instructions; displayed during payment entry when the account is used as a credit party.
DUEFR_ACC	(varchar 34). Due-from account at the remote bank.

Relationship File Tables

DBT_GL_RECON	(varchar 12). Debit general ledger reconciliation. Values are bank-defined in configuration files, determined from configuration routines, and used by back-end accounting systems for reconciliation.
CDT_GL_RECON	(varchar 12). Credit general ledger reconciliation. Values are bank-defined in configuration files, determined from configuration routines, and used by back-end accounting systems for reconciliation.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
STATEMENT_DATE	(varchar 8). Date of the account statement.
STATEMENT_NUMBER	(varchar 5). Number of the statement.
STATEMENT_SEQ_NO	(varchar 3). Statement sequence number.

Account Address Table (ACCOUNT_ADDRESS_T)

The Account Address table provides a cross reference to all accounts and addresses. This table links the Account (ACCOUNT_T) and the Address (ADDRESS_T) tables.

An address can have many accounts and an account can have many addresses (for example, a CHIPS universal identifier code, S.W.I.F.T. bank identifier code, or Telex answerback).

Table Relationships

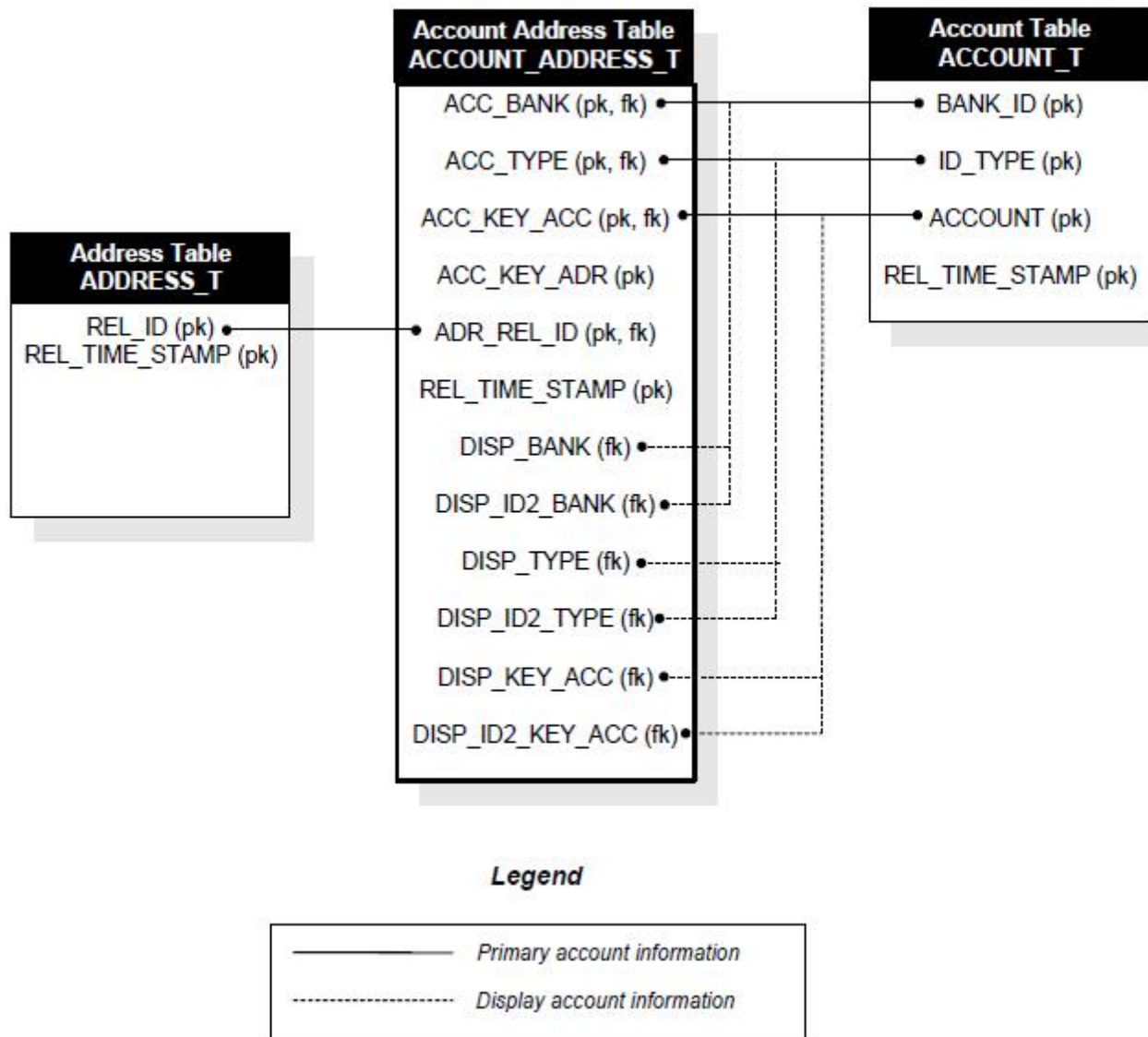
The following table lists the foreign keys in the Account Address table that relate to columns in the Account table:

These Account Address table columns...	Relate to these Account table columns...
ACC_BANK ACC_TYPE ACC_KEY_ACC	BANK_ID ID_TYPE ACCOUNT
DISP_BANK DISP_TYPE DISP_KEY_ACC	BANK_ID ID_TYPE ACCOUNT
DISP_ID2_BANK DISP_ID2_TYPE DISP_ID2_KEY_ACC	BANK_ID ID_TYPE ACCOUNT

The ADR_REL_ID column in the Account Address table relates to the REL_ID column in the Address table.

Relationship Diagram

This relationship diagram shows the Account Address table along with the Account and Address tables.



Column Descriptions

ACC_BANK (pk, fk)	(varchar 3). Bank identifier of the bank that owns the address record.
ACC_TYPE (pk, fk)	(varchar 1). Address ID type; contains the following values: A (ABA)

Relationship File Tables

	D (DDA) F (Foreign nostro account number) G (General ledger account number) P (CHIPS participant ID number) S (S.W.I.F.T.) V (Savings account number)
ACC_KEY_ACC (pk, fk)	(varchar 30). Account number used to find an address.
ACC_KEY_ADR (pk)	(varchar 5). Address code used to distinguish among multiple addresses sharing the same account.
ADR_REL_ID (pk, fk)	(number 10). Unique system-assigned reference number for this address.
REL_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
ACC_KEY_PAD	(varchar 1). Number of additional lines for account type addresses when an account has several addresses.
DISP_DEFAULT	(varchar 1). Flag indicating whether this account is the default account. This column contains the following values: X (default account) or blank or space (not default account). Operators use display columns to find addresses or to choose from ambiguous address identifiers.
DISP_NOF	(varchar 1). Flag indicating whether the account is not on file in REL. Contains the following values: * (account is not on file) , or blank or space (account is on file). Operators use display columns to find addresses or to choose from ambiguous address identifiers.
DISP_CUR	(varchar 3). Base currency of the account. Operators use the display currency column to choose the proper account associated with an address when more than one account is associated with an address.
DISP_BANK (fk)	(varchar 3). Account bank ID. Operators use the display bank column to choose from ambiguous address identifiers.
DISP_TYPE (fk)	(vvarchar 1). Account ID type; contains the following values: D (DDA) F (Foreign nostro account number) G (General ledger account number) P (CHIPS participant ID) V (Savings account number)
DISP_KEY_ACC (fk)	(varchar 30). Account number or account ID and address. Operators use display columns to find addresses or to choose from ambiguous address identifiers.
DISP_KEY_ADR	(varchar 5). Address code. Operators use display columns to find addresses or to choose from ambiguous address identifiers.

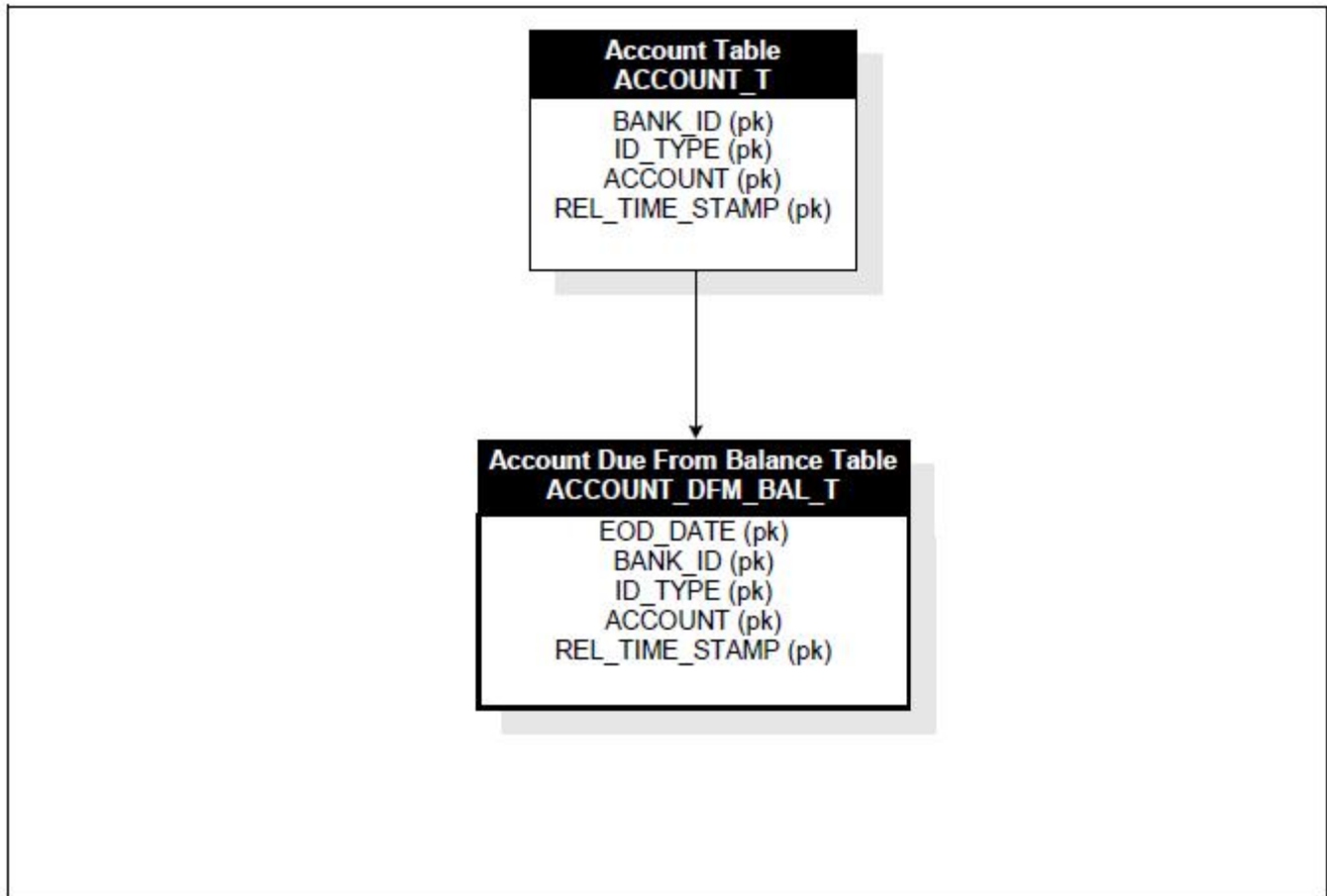
DISP_KEY_PAD	(varchar 1). Number of additional lines for account type addresses when an account has several addresses.
MEMO	(varchar 80). Descriptive text displayed in address account sequence.
DISP_ID2_BANK (fk)	(varchar 3). Bank ID.
DISP_ID2_TYPE (fk)	(varchar 1). Account type.
DISP_ID2_KEY_ACC (fk)	(varchar 30). Account number for account type address indices.
DISP_ID2_KEY_ADR	(varchar 5). Address code for account type address indices.
DISP_ID2_KEY_PAD	(varchar 1). Padding for account type address indices.
DISP_ADR1	(varchar 35). Temporary queue.
DISP_ADR2	(varchar 35). Temporary queue.
DISP_ADR3	(varchar 35). Temporary queue.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Account Due-From Balance Table (ACCOUNT_DFM_BAL_T)

The Account Due-From Balance table shows end-of-day balances for due-from accounts.

Table Relationships

This relationship diagram shows the Account Due-From Balance table along with its parent table, the Account table.



Column Descriptions

BANK_ID (pk)	(varchar 3). Bank ID of the bank that owns this address record; mapped from the Bank ID field in the Funds Transfer (FTR) screen.
ID_TYPE (pk)	(varchar 1). Account ID type of the due-from account; contains the following values: D (DDA) F (Foreign nostro account number) G (General ledger account number) V (Savings account number)
ACCOUNT (pk)	(varchar 30). Account number.

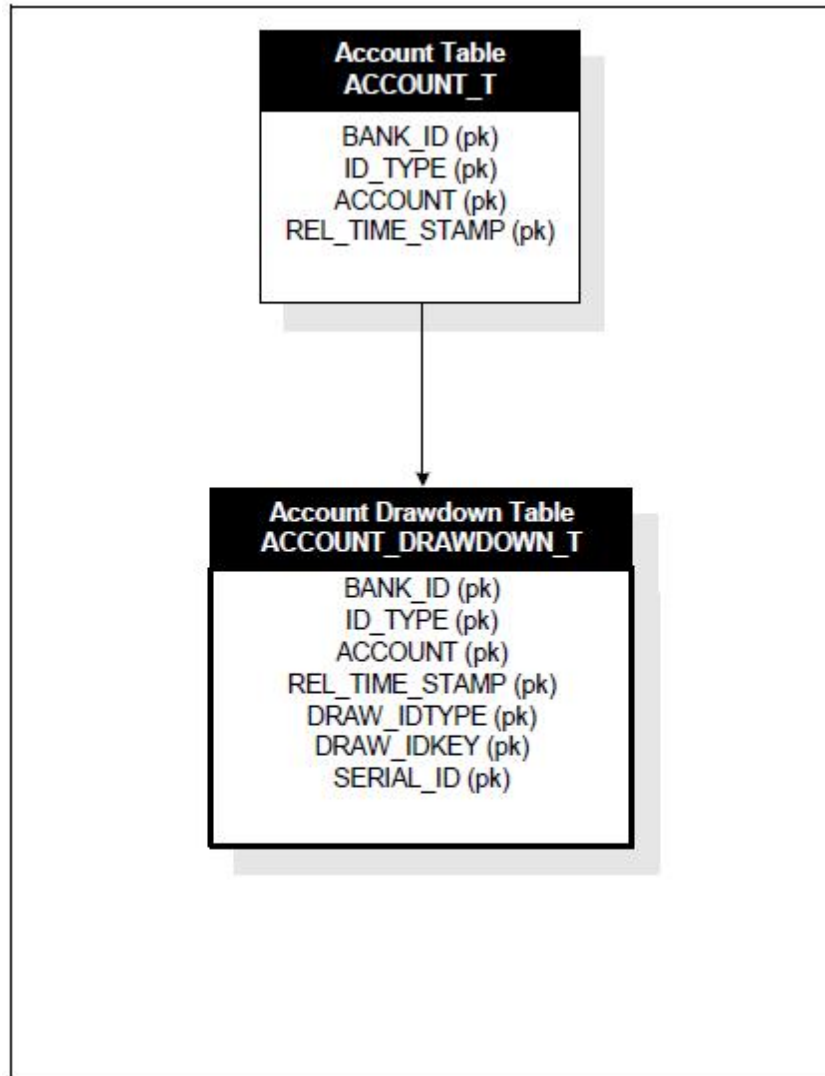
EOD_DATE (pk)	(date). Date of the end-of-day balance.
REL_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
EOD_BALANCE	(number 21.3). End-of-day account balance.
CHANGE_PERSON	(varchar 10). Operator ID and bank ID of the operator who changed the address.
CHANGE_COUNT	(number 5). Identifier to assure accuracy between end-of-day change queues; indicates the latest change level for each queue update.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Account Drawdown Table (ACCOUNT_DRAWDOWN_T)

The Account Drawdown table (ACCOUNT_DRAWDOWN_T) provides information regarding the ability of an account to draw down the balance of other accounts.

Table Relationships

This relationship diagram shows the Account Drawdown table along with its parent table, the Account table.



Column Descriptions

BANK_ID (pk)	(varchar 3). Bank that owns this account record.
ID_TYPE (pk)	(varchar 1). Account ID type; contains the following values: D (DDA) F (Foreign nostro account number) G (General ledger account number)

	P (CHIPS participant ID) V (Savings account number)
ACCOUNT (pk)	(varchar 30). Account number.
DRAW_IDTYPE (pk)	(varchar 1). Account ID type of the drawdown sender; the only possible value is A (ABA routing number).
DRAW_IDKEY (pk)	(varchar 64). The name, routing ID, or account number/address code of the drawdown.
SERIAL_ID (pk)	(number 19). A number that uniquely identifies a drawdown record.
REL_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
BNF_IDTYPE	(varchar 1). Account ID type of the drawdown beneficiary; contains the following values: C (CHIPS user ID) E (Extended ID)
BNF_IDKEY	(varchar 64). The name, routing ID, or account number/address code of the drawdown beneficiary.
DRAW_TRNSF_LIM	(number 21.3). Drawdown transfer limit.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
DRAW_TRNSF_LIM_CURRENCY	(varchar 3). Drawdown transfer limit currency.

Account Subaccount Table (ACCOUNT_SUB_ACC_T)

The Account Subaccount table contains information about subaccounts related to a concentration account.

When a subaccount is related to a concentration account, MTS uses the concentration account's balances when performing risk checking for the subaccount. Additionally, MTS uses the concentration account's pre-advise limit when performing pre-advise limit checking for the subaccount. This table is populated only if the CONCEN_ACCT_FLAG column in the Account table (ACCOUNT_T) contains a Y.

Table Relationships

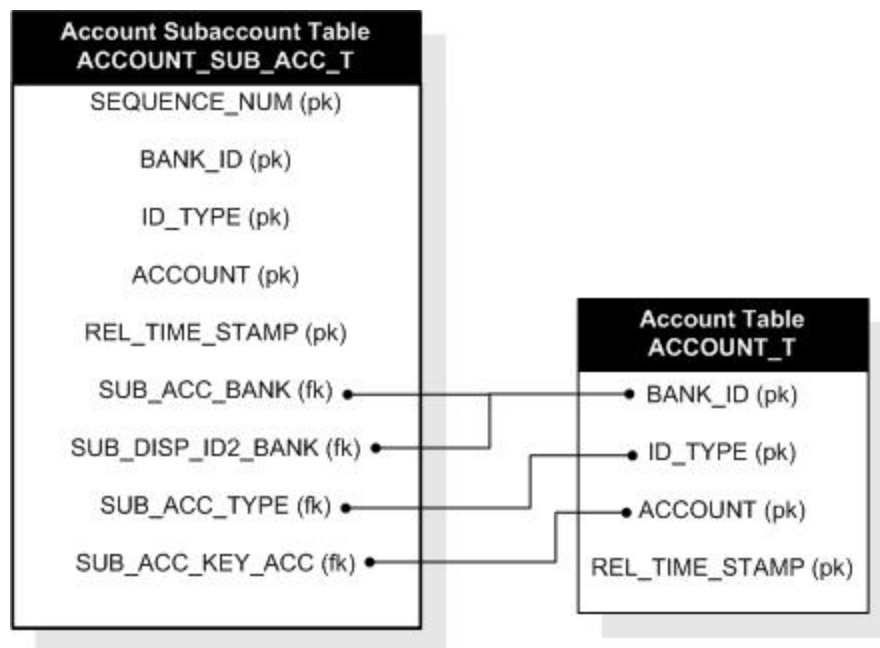
The following table lists the columns in the Account Subaccount table that relate to columns in the Account table:

Relationship File Tables

These Account Subaccount table columns...	Relate to these Account table columns...
SUB_ACC_BANK SUB_DISP_ID 2_BANK SUB_ACC_TYPE SUB_ACC_KEY_ACC	BANK_ID BANK_ID I D_TYPE ACCOUNT

Relationship Diagram

This relationship diagram shows the Account Subaccount table along with its parent table, the Account table.



Column Descriptions

Many tables in the MTS database are modeled on a common data structure. However, not all columns in the data structure are used in every table. As a result, some columns exist in this table, but are not populated. These column names are indicated by regular type (that is, not bold).

BANK_ID (pk)	(varchar 3) Bank identifier of the bank where this subaccount was created.
ID_TYPE (pk)	(varchar 1) Subaccount ID type; contains the following values: D (DDA) F (Foreign nostro account number) G (General ledger account number) V (Savings account number)

ACCOUNT (pk)	(varchar 30) Account number.
SEQUENCE_NUM (pk)	(number 10) A sequential number that uniquely identifies this subaccount in a sequence of subaccounts.
REL_TIME_STAMP (pk)	(varchar 16) Date and time when the record was updated.
SUB_ACC_BANK (fk)	(varchar 3) Bank identifier of the bank where this record was created.
SUB_ACC_TYPE (fk)	(varchar 1) Subaccount ID type; contains the following values: D (DDA) F (Foreign nostro account number) G (General ledger account number) V (Savings account number)
SUB_ACC_KEY_ACC (fk)	(varchar 30) Account number for subaccount addresses.
SUB_ACC_KEY_ADR	(varchar 5) Address code for subaccount addresses.
SUB_ACC_KEY_PAD	(varchar 1) Number of additional lines for account type addresses when an account has several addresses.
SUB_ADR_REL_ID	(number 10) REL ID of the subaccount.
SUB_DISP_DEFAULT	(varchar 1) Not populated in this table.
SUB_DISP_NOF	(varchar 1) Not populated in this table.
SUB_DISP_CUR	(varchar 3) Not populated in this table.
SUB_DISP_BANK	(varchar 3) Not populated in this table.
SUB_DISP_TYPE	(varchar 1) Not populated in this table.
SUB_DISP_KEY_ACC	(varchar 30) Not populated in this table.
SUB_DISP_KEY_ADR	(varchar 5) Not populated in this table.
SUB_DISP_KEY_PAD	(varchar 1) Not populated in this table.
SUB_MEMO	(varchar 80) Not populated in this table.
SUB_DISP_ID2_BANK (fk)	(varchar 3) Bank ID of the subaccount.
SUB_DISP_ID2_TYPE	(varchar 1) Subaccount ID type.
SUB_DISP_ID2_KEY_ACC	(varchar 30) Account number for account type address indices.
SUB_DISP_ID2_KEY_ADR	(varchar 5) Address code for account type address indices.

Relationship File Tables

SUB_DISP_ID2_KEY_PAD	(varchar 1) Padding for account type address indices.
SUB_DISP_ADR1	(varchar 35) Temporary queue.
SUB_DISP_ADR2	(varchar 35) Temporary queue.
SUB_DISP_ADR3	(varchar 35) Temporary queue.
RECORD_EXPIRED	(varchar 16) Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1) Indicates whether this row's timestamp column has been propagated to previous rows.

Account Group Table (ACCOUNT_GRP_T)

The Account Group table contains information about a group of accounts. This table is a parent table to the Account Group Member table (ACCOUNT_GRP_MBR_T).

Table Relationships

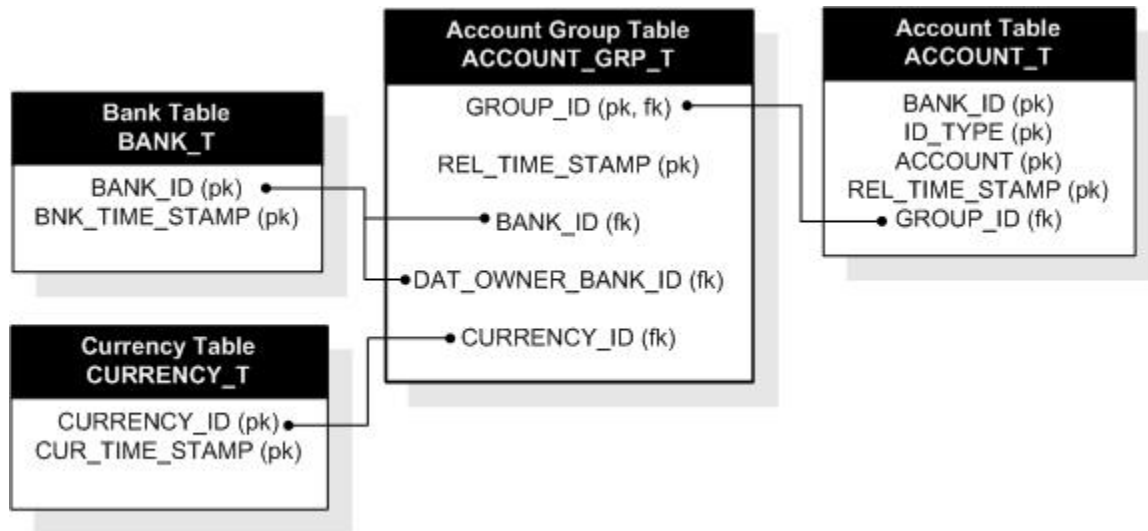
The GROUP_ID column in the Account Group table relates to the GROUP_ID column in the Account table.

The BANK_ID column and the DAT_OWNER_BANK_ID column in the Account Group table relate to the BANK_ID column in the Bank table:

The CURRENCY_ID column in the Account Group table relates to the CURRENCY_ID column in the Currency table.

Relationship Diagram

This relationship diagram shows the Account Group table and the tables related to it through foreign keys.



Column Descriptions

GROUP_ID (pk, fk)	(varchar 10). Group identification code.
REL_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
DAT_FUNCTION	(varchar 7). Function that last accessed this account record; contains the following values: VFY ADD (Verify add) VFY DEL (Verify delete) VFY UPD (Verify update)
CREATOR	(varchar 10). Operator ID and bank ID of the operator who created the group record.
CREATE_DATE	(date). Date that the account group record was created.
UPDATER	(varchar 10). Operator ID and bank ID of the operator who updated the account group record.
UPDATE_DATE	(date). Date that the account group record was updated.
VERIFIER	(varchar 10). Operator ID and bank ID of the operator who verified the account group record.

Relationship File Tables

VERIFY_DATE	(date). Date that the account group record was verified.
DAT_OWNER_PROD_ID	(varchar 3). Product ID.
DAT_OWNER_BANK_ID (fk)	(varchar 3). Bank ID.
DAT_OWNER_LOC	(varchar 6). Location ID.
DAT_OWNER_CUST	(varchar 20). Customer ID.
MINIMUM	(number 21.3). Minimum observed account group balance.
SOD_CHF	(number 21.3). Start-of-day account group balance at the clearing house.
SOD_LEDGER	(number 21.3). Start-of-day ledger balance for the account group.
SOD	(number 21.3). Start-of-day available account group balance.
BANK_ID (fk)	(varchar 3). Bank identifier of the bank that owns this group of accounts.
CURRENCY_ID (fk)	(varchar 3). Currency identification.
GROUP_NAME	(varchar 35). Group name.
GRP_DEPT	(varchar 20). Site-defined account group department column used for site-specific applications.
GRP_DEPT_NAME	(varchar 35). Site-defined account group department name used for site-specific applications.
GRP_OFF_EMP_ID	(varchar 10). Employee ID of the bank officer responsible for the account group.
GRP_OFF_PHONE	(varchar 20). Phone number of the bank officer responsible for the account group.
GRP_OFF_DEPT	(varchar 20). Department of the bank officer responsible for the account group.
GRP_OFF_NAME	(varchar 25). Name of the bank officer responsible for the account group.
ALT_OFF_EMP_ID	(varchar 10). Employee ID of the alternate bank officer responsible for the account group.
ALT_OFF_PHONE	(varchar 20). Phone number of the alternate bank officer responsible for the account group.
ALT_OFF_DEPT	(varchar 20). Department of the alternate bank officer responsible for the account group.
ALT_OFF_NAME	(varchar 25). Name of the alternate bank officer responsible for the account group.

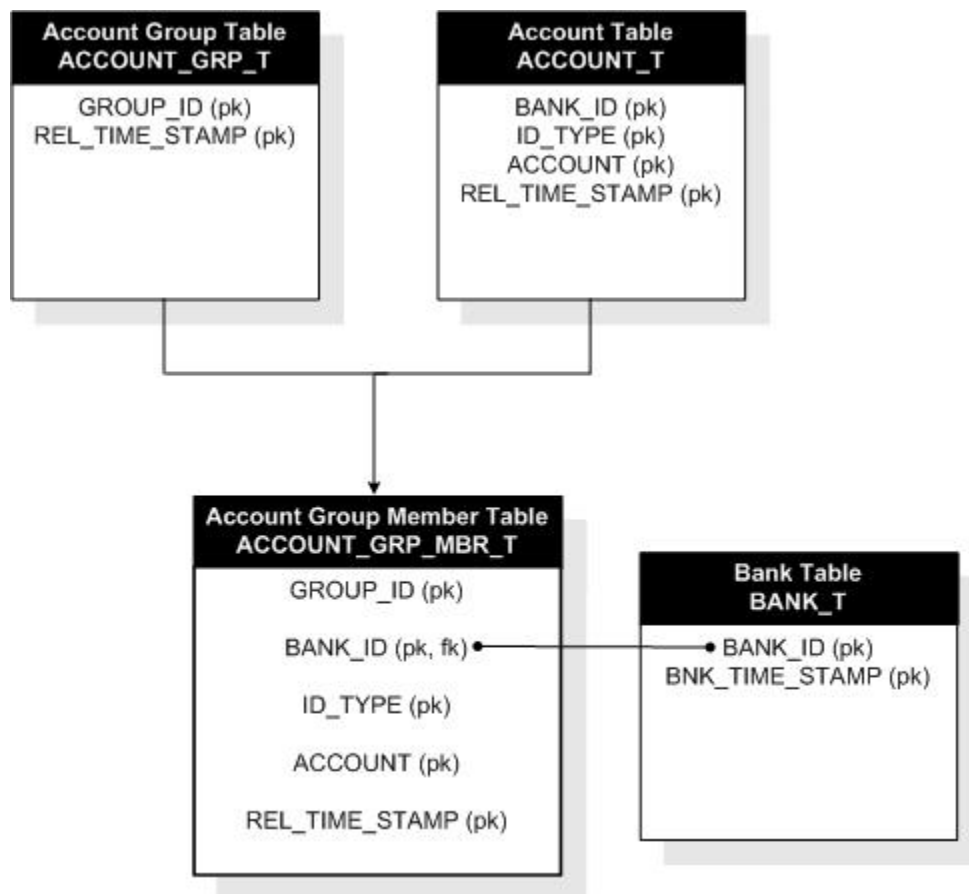
CONC_ACCT	(varchar 30). Not populated for account groups.
CREDIT_LIMIT	(number 21.3). Credit limit for the group.
DAYLIGHT_OD	(number 21.3). Daylight overdraft amount for the account group.
DAYLIGHT_OD_EXP	(date). Daylight overdraft expiration date for the account group.
PREADV_LIMIT	(number 21.3). Preadvise limit for the group.
PREADV_LIMIT_EXP	(date). Preadvise limit expiration date.
TARGET_PERIOD	(varchar 1). Period for the target balance set by a bank on their customers' accounts. This column contains the following values: D (Day) W (Week) F (Fortnight or two weeks) M (Month) Y (Year)
TARGET_BALANCE	(number 21.3). Target account balance set by a bank on their customers' accounts; shown in account position reports.
TXN_LIMIT_CUR	(varchar 15). Transaction limit for transactions in a specific currency.
TXN_LIMIT	(number 21.3). Transaction limit for the group.
NON_RPTV_LIMIT	(number 21.3). Non-repetitive callback limit.
RPTV_LIMIT	(number 21.3). Repetitive callback limit.
DLY_RLS_HRS	(varchar 2). Hour portion of the time until which MTS holds Auto-hold payments.
DLY_RLS_MNS	(varchar 2). Minute portion of the time until which MTS holds Auto-hold payments.
DBT_HOLD_FLAG	(varchar 1). Flag (set in the Relationship File). indicating whether MTS sends the group's debit transactions to risk.
CDT_HOLD_FLAG	(varchar 1). Flag (set in the relationship File). indicating whether MTS sends the group's credit transactions to risk.
RISK_SPEC_INSTR1	(varchar 35). First line of risk special instructions.
RISK_SPEC_INSTR2	(varchar 35). Second line of risk special instructions.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Account Group Member Table (ACCOUNT_GRP_MBR_T)

The Account Group Member table contains information about members in an account group. Account group members can be individual accounts or another account group.

Table Relationships

This relationship diagram shows the Account Group Member table, its two parent tables, the Account table and the Account Group table, and the Bank table.



Column Descriptions

GROUP_ID (pk)	(varchar 10). Group identification code.
BANK_ID (pk, fk)	(varchar 3). Bank identifier of the bank that owns this account.
ID_TYPE (pk)	(varchar 1). Account ID type; contains the following values:

	D (DDA) F (Foreign nostro account number) G (General ledger account number) V (Savings account number) 1 (Group ID)
ACCOUNT (pk)	(varchar 30). Account number or group ID, if ID_TYPE is 1.
REL_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
ACC_NAME	(varchar 35). Account name.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Address Tables

The address tables contain information about the behavior of the accounts related to an address including advice methods, destinations, PINs, standing instructions, and test history.

Parent/child relationships

The Address table (ADDRESS_T) is the parent table to the following child tables:

- Address Destination (ADDRESS_DEST_T)
- Address Test History (ADDRESS_TST_HST_T)
- Address PIN Privilege (ADDRESS_PIN_PRIV_T)

Primary keys

All address tables have these two primary keys in common:

- REL_ID
- REL_TIME_STAMP

Address Table (ADDRESS_T)

The Address table contains address information for the accounts including the relationship type, routing codes, and phone numbers.

Table Relationships

The REL_ID column in the Address table relates to the ADR_REL_ID column in the Account Address table.

The following columns in the Address table relate to the ALGORITHM_ID column in the Algorithm table:

- IN_PREV_ALG_ID
- IN_CUR_ALG_ID
- IN_NEXT_ALG_ID
- OUT_PREV_ALG_ID
- OUT_CUR_ALG_ID
- OUT_NEXT_ALG_ID

The following columns in the Address table relate to the BANK_ID column in the Bank table:

- BANK_ID
- DAT_OWNER_BANK_ID

The ADR_CDT_DEPT and ADR_DBT_DEPT columns in the Address table relate to the DEPT_ID column in the Department table.

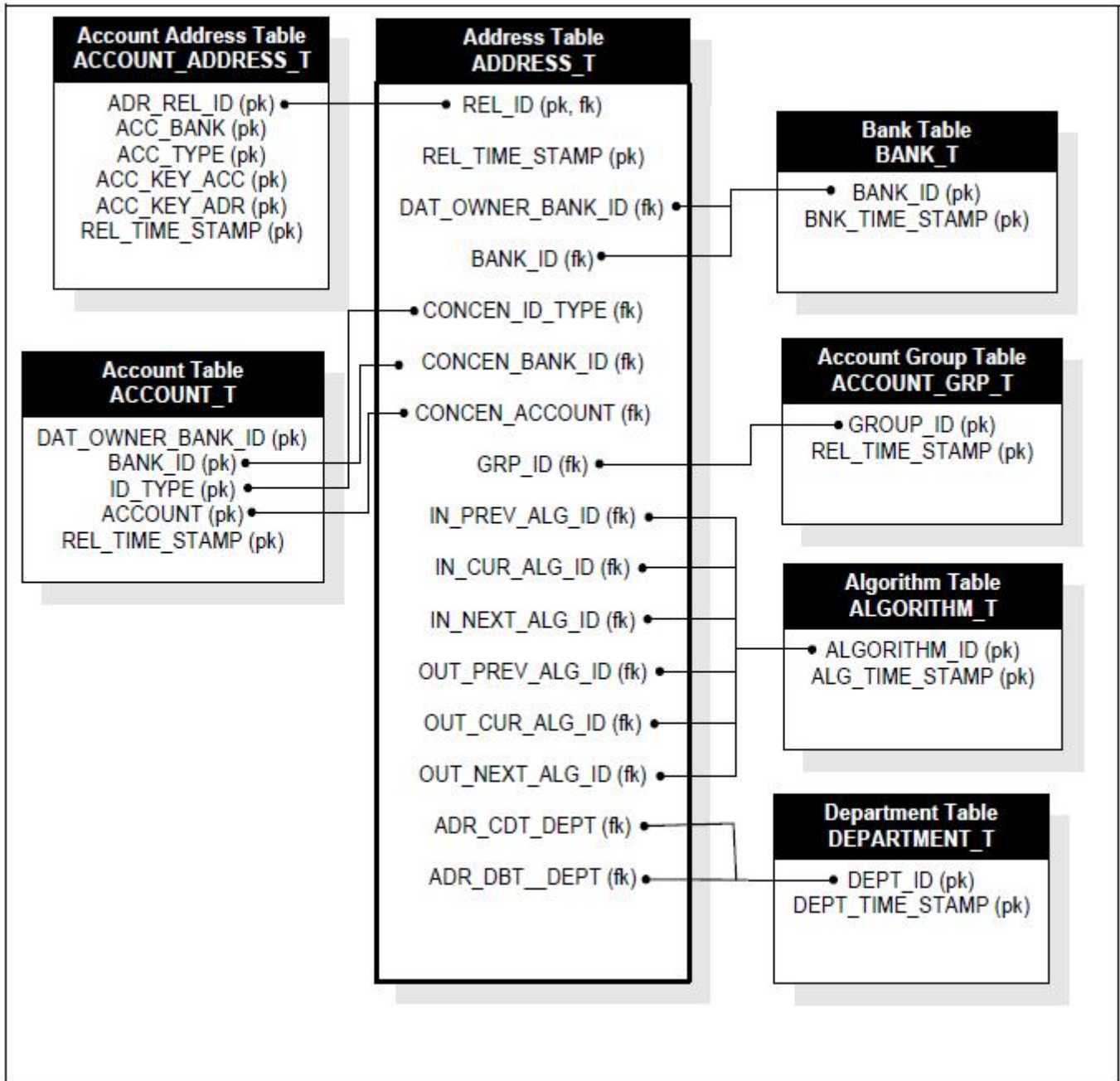
The GRP_ID column in the Address table relates to the GRP_ID column in the Account Group table.

The following table lists the columns in the Address table that relate to columns in the Account table:

These Address table columns...	Relate to these Account table columns...
CONCEN_ACCOUNT	ACCOUNT
CONCEN_BANK_ID	BANK_ID
CONCEN_ID_TYPE	ID_TYPE

Relationship Diagram

This relationship diagram shows the Address table along with the tables that are related to it through foreign keys.



Column Descriptions

REL_ID (pk, fk)	(number 10). Unique system-assigned reference number for this address.
REL_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
DAT_FUNCTION	(varchar 7). Function that last accessed this account record; contains the following values: VFY ADD (Verify add) VFY DEL (Verify delete) VFY UPD (Verify update)
CREATOR	(varchar 10). Operator ID and bank ID of the operator who created the address record.
CREATE_DATE	(date). Date that the address record was created.
UPDATER	(varchar 10). Operator ID and bank ID of the operator who updated the address record.
UPDATE_DATE	(date). Date that the address record was updated.
VERIFIER	(varchar 10). Operator ID and bank ID of the operator who verified the address record.
VERIFY_DATE	(date). Date that the address record was verified.
DAT_OWNER_PROD_ID	(varchar 3). Product ID.
DAT_OWNER_BANK_ID (fk)	(varchar 3). Bank ID.
DAT_OWNER_LOC	(varchar 6). Location ID.
DAT_OWNER_CUST	(varchar 20). Customer ID.
LAST_USED	(date). Last date on which this address was used.
SPECIAL_IDBANK	(varchar 3). Bank ID.
SPECIAL_IDKEY	(varchar 20). Generic bank name.
CONCEN_BANK_ID (fk)	(varchar 3). The bank ID at which the concentration account resides in a multibanking environment.
CONCEN_ID_TYPE (fk)	(varchar 1). The account ID type of the concentration account. Valid values are: D (Demand Deposit) G (General Ledger) F (Foreign Nostro) P (CHIPS Participant)

Relationship File Tables

	V (Savings)
CONCEN_ACCOUNT (fk)	(varchar 30). Account number of the concentration account.
GRP_ID (fk)	(varchar 10). Group ID number.
WE_INHIB_THEM	(varchar 1). CHIPS bilateral information.
THEY_INHIB_US	(varchar 1). CHIPS bilateral information.
ERROR_INHIB	(varchar 1). Indicates whether a CHIPS bilateral information error exists.
CHP_RCV_LIMIT	(number 21.3). Incoming CHIPS message limit. Messages exceeding this amount go to risk.
CHP_SND_LIMIT	(number 21.3). Outgoing CHIPS message limit. Messages exceeding this amount go to risk.
ACC_MINIMUM	(number 21.3). Minimum observed account balance at the start-of-day.
ACC_SOD_CHF	(number 21.3). Start-of-day clearinghouse funds balance.
ACC_SOD_LEDGER	(number 21.3). Start-of-day ledger balance for an account.
ACC_SOD	(number 21.3). Start-of-day account balance.
GRP_MINIMUM	(number 21.3). Minimum observed group balance.
GRP_SOD_CHF	(number 21.3). Start-of-day clearinghouse funds balance for a group.
GRP_SOD_LEDGER	(number 21.3). Start-of-day ledger balance for a group.
GRP_SOD	(number 21.3). Start-of-day balance for a group.
BANK_ID (fk)	(varchar 3). Bank identifier of the bank that owns this address.
REL_TYPE	(varchar 12). Indicates the relationship file record type: 1 (Account) 2 (Address) 3 (Group) 4 (Auxiliary)
ADDRESS_ID	(varchar 5). Address identifier appended to the account number used to distinguish between multiple addresses sharing the same account ID.
ADR_TYPE_CODE	(varchar 1). Address type; mapped from the Relationship File (REL). This column contains the following values: B (Bank) I (Internal)

	C (Corporate)
ADR_SUB_TYPE	(varchar 3). Bank sub-type code; usually has a two-letter country code as a prefix and is available for DDA post to send to mainframe systems. This column contains the following values: COM (Commercial) SAV (Savings bank) S+L (Savings + loan) COP (Cooperative) C/U (Credit union) EDG (Edge Act) F/A (Foreign agency) F/B (Foreign bank) AFF (Affiliate) IBF (Internal banking facility) CEN (Central) DOM (Domestic) IND (International domestic) INF (International foreign) OTH (Other)
SNAME_ID	(varchar 30). Short name identifier associated with the address.
ABA_ID	(varchar 9). American Bankers Association routing number.
SWIFT_ID	(varchar 30). 8- or 11-character S.W.I.F.T. bank identifier code.
BRANCH_ID	(varchar 30). Branch identification number.
CHIPS_UID_ID	(varchar 30). CHIPS universal identifier code.
USER_ID	(varchar 30). User-supplied identification number for the address.
INTERBNK_ID	(varchar 30). Interbank identification number (supplied by the bank) .
CUSTOMER_ID	(varchar 30). Customer ID for the current address. MTS uses this value to retrieve address information from the customer index.
CUSTOMER_NAME	(varchar 35). Customer name for the current address.
ADR_NAME	(varchar 35). Address name.
ADR1	(varchar 35). First line of mailing address.
ADR2	(varchar 35). Second line of mailing address.
ADR3	(varchar 35). Third line of mailing address.

Relationship File Tables

ZIP	(varchar 10). Customer's mailing postal code.
COUNTRY_CODE	(varchar 2). Customer's address country code.
PHN	(varchar 20). Customer's phone number.
EXT	(varchar 4). Customer's extension.
SPEED	(varchar 4). Customer's speed dial number.
ADR_DESC	(varchar 20). Address description used to help identify the correct address; mapped from the Relationship File (REL) Address File screen.
ADR_CLASS	(varchar 6). Address class column used to help identify correct addresses and for site-specific applications; mapped from the Relationship File (REL) Address File screen.
ADR_CITY	(varchar 15). Address city column used to help identify correct addresses and for site-specific applications; mapped from the Relationship File (REL) Address File screen.
ADR_DBT_DEPT (fk)	(varchar 20). Address debit department column used for site-specific applications; mapped from the Relationship File (REL) Address File screen.
ADR_CDT_DEPT (fk)	(varchar 20). Address credit department column used for site-specific applications; mapped from the Relationship File (REL) Address File screen.
ADR_LOCATION	(varchar 6). Address location.
DBT_RT_CODE	(varchar 6). Debit routing code for confirmations that identifies a named group of printers.
CDT_RT_CODE	(varchar 6). Credit routing code for confirmations that identifies a named group of printers.
CUSTOMER_CODE	(varchar 10). Bank-defined code for confirmations that identifies the customer for the current address.
FED_SHORT_NAME	(varchar 20). Federal Reserve short name.
FED_ONLINE_FLAG	(varchar 1). Federal Reserve online status flag; contains the following values: Y (Online) N (Not online) F (Foreign) I (Not eligible) S (Settlement)
DBT_SPEC_INSTR1	(varchar 35). First line of debit special instructions.
DBT_SPEC_INSTR2	(varchar 35). Second line of debit special instructions.

DBT_SPEC_INSTR3	(varchar 35). Third line of debit special instructions.
CDT_SPEC_INSTR1	(varchar 35). First line of credit special instructions.
CDT_SPEC_INSTR2	(varchar 35). Second line of credit special instructions.
CDT_SPEC_INSTR3	(varchar 35). Third line of credit special instructions.
PHONE_TEST_TYPE	(varchar 1). Authentication method used for phone advising; the bank defines up to 8 tests in configuration files. This column contains the following values: 1 – 8 (ID number of bank test defined in configuration files) T (Use telex testword authentication)
PHONE_EARLY_HRS	(varchar 2). Not populated in this table.
PHONE_EARLY_MNS	(varchar 2). Not populated in this table.
DOCUMENT_ID	(varchar 6). Change control document ID; tracks the last change to this address.
RISK_COUNTRY	(varchar 2). Country code of the risk country (a bank's headquarters is located in its risk country).
DEBIT_AUTH_FLAG	(varchar 1). Debit authority flag that determines whether this address has authority to debit all accounts. This column contains the following values: Y (Authority to debit all accounts) N (No authority to debit accounts)
LOCAL_FLAG	(varchar 1). Flag indicating whether an address is local. This column contains the following values: L (Local; message travels through a secure link) N (Non local; message travels through a non-secure link)
PRIORITY_FLAG	(varchar 1). Priority flag used for informational, site-specific purposes only.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
TST_PARENT_REL_ID	(number 10). Reference number of the parent address used for testword authentication.
TST_CONTACT	(varchar 20). Name of the person who performs testword authentication.
TEST_PHONE_EXT	(varchar 4). Phone extension of the person who performs testword authentication.
TEST_PHONE_NUM	(varchar 12). Phone number of the person who performs testword authentication.
TST_INSTR1	(varchar 35). Test instruction 1.

Relationship File Tables

TST_INSTR2	(varchar 35). Test instruction 2.
DORMANT_FLAG	(varchar 1). Flag indicating whether the testword algorithm for this address is active or dormant. This column contains the following values: blank (active) or Y (dormant).
SLOTS_SAVED	(number 5). Number of slots of testword history to save.
SERIAL_RESET_FLAG	(varchar 1). Flag that causes the testword serial table to be reset to 1 when the first message of a new day is authenticated.
GAP_BO_LIMIT	(number 5). Number of slots to search backward from the current end of the testword history for a usable gap when the operator is using the FAI or OFL function.
GAP_BS_LIMIT	(number 5). Number of slots to search backward from the current end of the testword history for a usable gap when the supervisor is using the FAI or OFL function.
GAP_BA_LIMIT	(number 5). Number of slots to search backward from the current end of the testword history for a usable gap when the autotester is using the FAI or OFL function.
GAP_FO_LIMIT	(number 5). Number of slots to search forward from the current end of the testword history for a usable gap when the operator is using the FAI or OFL function.
GAP_FS_LIMIT	(number 5). Number of slots to search forward from the current end of the testword history for a usable gap when the supervisor is using the FAI or OFL function.
GAP_FA_LIMIT	(number 5). Number of slots to search forward from the current end of the testword history for a usable gap when the autotester is using the FAI or OFL function.
IN_GENERATE_FLAG	(varchar 1). Flag indicating whether the incoming message testword algorithm is generated by the system. Y (system). or blank (manually). can appear in this field.
IN_ACK_FIX_DATE	(date). Date when the next fixed number used to create the incoming testword algorithm was acknowledged. If this column is blank, the fixed number has not been acknowledged.
IN_FIX_PERSON	(varchar 10). Operator ID and bank ID of the operator who changed the fixed number used to create the incoming testword algorithm.
IN_PREV_ALG_ID (fk)	(varchar 12). Previous incoming message testword algorithm.
IN_PREV_FIX_DATE	(date). Effective date of previous incoming message testword algorithm.
IN_CUR_ALG_ID (fk)	(varchar 12). Current incoming message algorithm.
IN_CUR_FIX_DATE	(date). Effective date of the current fixed number used to create the incoming message testword algorithm.
IN_NEXT_ALG_ID (fk)	(varchar 12). Next incoming message testword algorithm.

IN_NEXT_FIX_DATE	(date). Effective date of the next fixed number used to create the incoming message testword algorithm.
OUT_GENERATE_FLAG	(varchar 1). Flag indicating whether the outgoing message testword algorithm is generated by the system. This column contains the following values: Y (system) or blank (manually).
OUT_ACK_FIX_DATE	(date). Date when the next fixed number used to create the outgoing testword algorithm was acknowledged. If this column is blank, the fixed number has not been acknowledged.
OUT_FIX_PERSON	(varchar 10). Operator ID and bank ID of the operator who changed the fixed number used to create the outgoing testword algorithm.
OUT_PREV_ALG_ID (fk)	(varchar 12). Previous outgoing message testword algorithm.
OUT_PREV_FIX_DATE	(date). Effective date of the previous outgoing testword algorithm.
OUT_CUR_ALG_ID (fk)	(varchar 12). Current outgoing message testword algorithm.
OUT_CUR_FIX_DATE	(date). Effective date of the current fixed number used to create the outgoing message testword algorithm.
OUT_NEXT_ALG_ID (fk)	(varchar 12). Next outgoing message testword algorithm.
OUT_NEXT_FIX_DATE	(date). Effective date of the next fixed number used to create the outgoing message testword algorithm.
SEC_ONLINE_FLAG	(varchar 1). Federal Reserve online status flag for ABAs only; contains the following values: Y (Online) N (Not online) F (Foreign) I (Not eligible) S (Settlement)
PENDING_DEL_DATE	(date). Date that the auxiliary address record was deleted. A deleted address record remains in the database until the period is purged.
CHIPS_OPEN_HH	(varchar 2). Hour portion of the time when CHIPS processing opens.
CHIPS_OPEN_MM	(varchar 2). Minute portion of the time when CHIPS processing opens.
FED_OPEN_HH	(varchar 2). Hour portion of the time when FED processing opens.
FED_OPEN_MM	(varchar 2). Minute portion of the time when FED processing opens.
TIME_ZONE_NAME	(varchar 10). Name of the time zone.
SRV_MSG_FLAG	(varchar 1). Flag indicating whether to send a service message, that is, notification, to the third party.

Relationship File Tables

INC_HDR_FLAG	(varchar 1). Flag indicating whether to include a line of header text describing the third party test.
CHIPS_ID	(varchar 30). Clearing House ID.
SWF_SUBTYPE	(varchar 4). SWIFT participant subtype.
PROFILE_IDBANK	(varchar 3). Differentiator to allow configurable cross bank indices.
PROFILE_IDTYPE	(char 1). ID type.
PROFILE_IDACC	(varchar 30). The account number for account type address indices.
PROFILE_IDADR	(varchar 5). The address code for account type address indices.
PROFILE_IDPAD	(char 1). Padding for account type address indices.

Address Destination Table (ADDRESS_DEST_T)

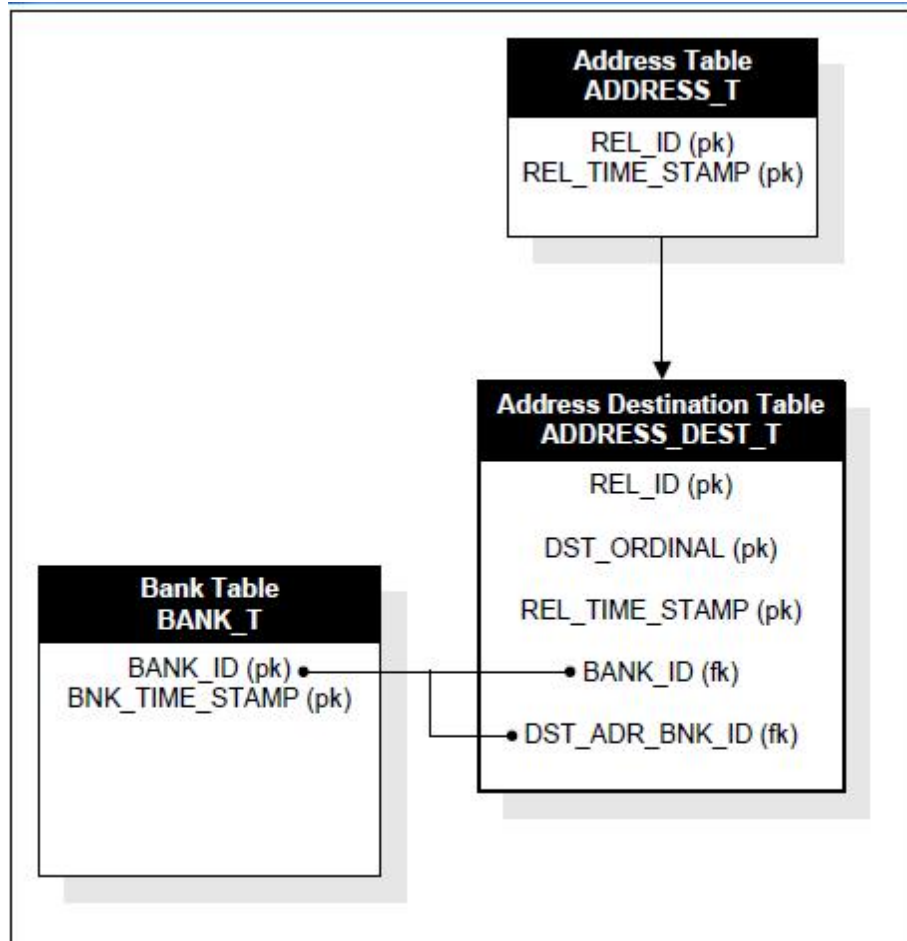
The Address Destination table provides information about destinations for accounts.

Table Relationships

The BANK_ID and DST_ADR_BNK_ID columns in the Address Destination table relate to the BANK_ID column in the Bank table.

Relationship Diagram

This relationship diagram shows the Address Destination table along with its parent table, the Address table, and the tables that are related to it through foreign keys.



Column Descriptions

REL_ID (pk)	(number 10). Unique system-assigned reference number for this address.
DST_ORDINAL (pk)	(number 5). Order of this destination within the sequence of destinations for this message.
REL_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
BANK_ID (fk)	(varchar 3). Bank ID of the bank to which the address belongs.
ADDRESS_ID	(varchar 5). Address identifier appended to the account number. This identifier distinguishes among multiple addresses sharing the same account ID.

Relationship File Tables

DST_ADR_BNK_ID (fk)	(varchar 3). Bank ID of the bank that owns the record found in the REL file.
DST_REL_ID	(number 10). Rel ID of the destination address.
DST_OVR	(varchar 1). Override flag; an asterisk(*) indicates this address is not on file. If the address is not on file, the MTS auxiliary file may have been used to determine the address.
DST_IDTYPE	(varchar 1). ID type of the destination account. This column contains the following values: A (Fedwire routing ABA number) B (Branch code) C (CHIPS user ID) K (Customer ID number) M (Private leased line network MAC code) S (8- or 11-character S.W.I.F.T. address) T (Telex number) U (User-defined name) X (Telex answerback) Z (Cable address)
DST_ID	(varchar 64). Destination ID; standard address key used to find this address in the REL file.
DST_TYPE	(varchar 3). Primary wire service/delivery type. This column contains the following values: ARX CHP (CHIPS) ENQ (Other processing queue) FAX (Fascimile) FED (Fedwire) ITX (Incoming telex) MAC (Machine authenticated code) PRT (Printer) PVT (Private carrier) SAF (Telex store and forward) SWF (S.W.I.F.T.) TLX (Telex) WTX (Outgoing telex)
DST_CARRIER	(varchar 3). Destination carrier types; contains the following values: CAB (Cable) FAX (Fax server) ITX (Incoming telex)

	MGM PVT (Private wire service) SAF (Telex store and forward) TLX (Telex) WTX (Outgoing telex)
DST_AREA_CODE	(varchar 4). Destination area code.
DST_DIAL	(vvarchar 16). Destination telephone or telex number. This column is used for the following destination types only: FAX (Fascimile) ITX (International telex) SAF (Telex store and forward) TLX (Telex) WTX (Outgoing telex)
DST_SIGN	(varchar 1). Identifies MAC (machine authenticated code) or PVT (private) carriers.
DST_MAC	(varchar 10). Code used for MAC (machine authenticated code) deliveries. This column is used for ARX and WTX destination types only.
DST_ANS	(vvarchar 20). Answerback character string used for telex deliveries. This column is used for the following destination types only: ITX (Incoming telex) SAF (Telex store and forward) TLX (Telex) WTX (Outgoing telex)
DST_NAME1	(varchar 35). First line of the correspondent or cable name and address. This column is used for CAB, MGM, and PVT destination carrier types only.
DST_CAB2	(varchar 35). Second line of the correspondent or cable name and address.
DST_CAB3	(varchar 35). Third line of the correspondent or cable name and address.
DST_CAB4	(varchar 35). Fourth line of the correspondent or cable name and address.
DST_CITY	(varchar 15). City of the correspondent or cable name and address.
DST_COUNTRY	(varchar 2). Country of the correspondent or cable name and address.
DST_QUEUE_PROD_ID	(varchar 3). MTS product ID (product part of a three-part queue name). . This column is used for ENQ and PRT destination types only.
DST_QUEUE_BANK_ID	(varchar 3). Owning bank ID (bank part of a three-part queue name). . This column is used for ENQ and PRT destination types only.

Relationship File Tables

DST_QUE_LOC	(varchar 6). Site-defined location (within the bank) where this message was created (location part of a three-part queue name). . MTRANS is usually the standard value. No location is associated with incoming network transactions. This column is used for ENQ and PRT destination types only.
DST_QUEUE_CUST	(varchar 20). MTS customer name (customer part of a three-part queue name). This column is used for ENQ and PRT destination types only.
DST_QUEUE_NAME	(varchar 33). MTS queue name (queue part of a three-part queue name). This column is used for ENQ and PRT destination types only.
DST_ROUTE_ID	(varchar 64). Appropriate routing ID when delivery is via FED, S.W.I.F.T, or PVT (private). This column contains the following values: ABA routing number, or a S.W.I.F.T. ID. This column is used for PVT and SWF destination types only.
DST_TESTKEY	(varchar 16). Generated code to authenticate an outgoing telex message.
DST_DEPARTMENT	(varchar 20). Department of correspondent if the destination is a printer queue name.
DST_RT_CODE	(varchar 6). Routing code that specifies a pre-defined printer group when delivery is via printer.
DST_ATTN	(varchar 60). Attention line; mapped from ATTN column on WRP (wire repair), FAX, and other screens.
DST_FORMAT	(varchar 4). Indicates the form name or letterhead form name used when creating a fax. Reserved for custom use.
DST_MSGTYPE	(varchar 4). Identifies the message type sent to this destination.
DST_ADR_LOCATION	(varchar 6). Site-defined printer location.
DST_SEC_TYPE	(varchar 3). Secondary or alternate wire service/delivery type; cannot be the same carrier as the primary delivery.
DST_SEC_AREA_CODE	(varchar 4). Secondary area code.
DST_SEC_DIAL	(varchar 16). Secondary telephone or telex number.
DST_SEC_SIGN	(varchar 1). Secondary column used for MAC (machine authenticated code) or PVT (private) carriers.
DST_SEC_MAC	(varchar 10). Secondary code used for MAC deliveries.
DST_SEC_ANS	(varchar 20). Secondary answerback code used for telex deliveries.
DST_SEC_NAME1	(varchar 35). First line of secondary correspondent or cable name and address.
DST_SEC_CAB2	(varchar 35). Second line of secondary correspondent or cable name and address.
DST_SEC_CAB3	(varchar 35). Third line of secondary correspondent or cable name and address.

DST_SEC_CAB4	(varchar 35). Fourth line of secondary correspondent or cable name and address.
MSG_COND	(varchar 3). Message type match criterion for the destination; contains the following values: EQL (equals the transaction's message type, route the message to the destination) or NEQ (does not equal the transaction's message type, route to the destination).
MSG_TYPE	(varchar 4). Federal Reserve outgoing message type.
SRC_COND	(varchar 3). Source match criterion for the destination; contains the following values: EQL (if the source equals the message source, perform the operation) or NEQ (if the source does not equal the message source, perform the operation).
SRC_ID	<p>(varchar 3). Transaction source code; contains the following values:</p> <p>ADJ (Adjustment)</p> <p>ADM (Administrative message entry function)</p> <p>CHP (CHIPS)</p> <p>CMS (Cash management system)</p> <p>DFM (Due-from monitor)</p> <p>DOC (Letter of credit)</p> <p>ENT (Payment entry function)</p> <p>ETW (Enhanced Treasury Workstation)</p> <p>FED (Fedwire)</p> <p>INT (Internal transfer)</p> <p>LTC (Letter requiring callback)</p> <p>LTR (Letter)</p> <p>MEM (Memo post function)</p> <p>MIS (Miscellaneous, for example: cash management system or remote batch entry)</p> <p>MTW (Money Transfer Workstation)</p> <p>OFL (Offline)</p> <p>PHN (Phone transfer initiated outside bank)</p> <p>RBE (Remote batch entry)</p> <p>SOD (Start-of-day balance load)</p> <p>STO (Standing order function)</p> <p>SWF (S.W.I.F.T.)</p> <p>WIR (Telex)</p> <p>User-defined code</p> <p>For example, you can use this column to route confirmations by source.</p>
DST_ORIGINAL_ORD	(number 5). Original serial number of this destination on the address.
DST_CHAR_COUNT	(number 10). Numeric count of characters in the message text stored in the destination set. This text varies according to the delivery mechanism. (This column is not currently implemented.).

Relationship File Tables

DST_MULTI_SEQ	(number 5). Reserved for future use.
DST_MULTI_START	(number 5). Reserved for future use.
DST_MULTI_COUNT	(number 5). Reserved for future use.
DST_OUT_TEST	(varchar 1). Flag indicating whether an outgoing testword is required (Y) or not (N).
DST_PDM_FLAG	(varchar 1). Flag indicating whether this delivery is known to be a possible duplicate (Y) or not (N).
DST_STATE	(varchar 1). Indicates whether message was delivered; contains the following values: D (Delivered) Blank (Not processed) Q (Queued for delivery) S (Selected for delivery) F (Selected and formatted for delivery) N (Instruction not acknowledged (NAK'd))
DST_ACK_STATE	(varchar 1). Reserved for future use.
DST_NAK_STATE	(varchar 1). Reserved for future use.
DST_DELIVERY_FLG	(varchar 1). Reserved for future use.
DST_TIME	(date). Destination transmission time.
DST_OUTGOING_REF	(varchar 80). Reference memo consisting of an outgoing serial number for a FED or CHIPS transaction.
DST_3RD_PARTY_TST	(varchar 1). Destination column that contains 3rd-party test information.
DST_TEST_DATE	(date). Date used to generate the testword for authenticating a telex message.
DST_FIN_COPY_ID	(varchar 3). Reserved for future use.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
DST_119_HDR_DATA	(varchar 80). Text contained in SWF header line 3, field 119.
DST_CASC_NAME	(varchar 33). Cascade routing.
DST_SRV_MSG_FLAG	(varchar 1). Flag indicating whether to send a service message, that is, notification, to the third party in this destination set.
DST_INC_HDR_FLAG	(varchar 1). Flag indicating whether to include a line of descriptive header text.

Address PIN Privilege Table (ADDRESS_PIN_PRIV_T)

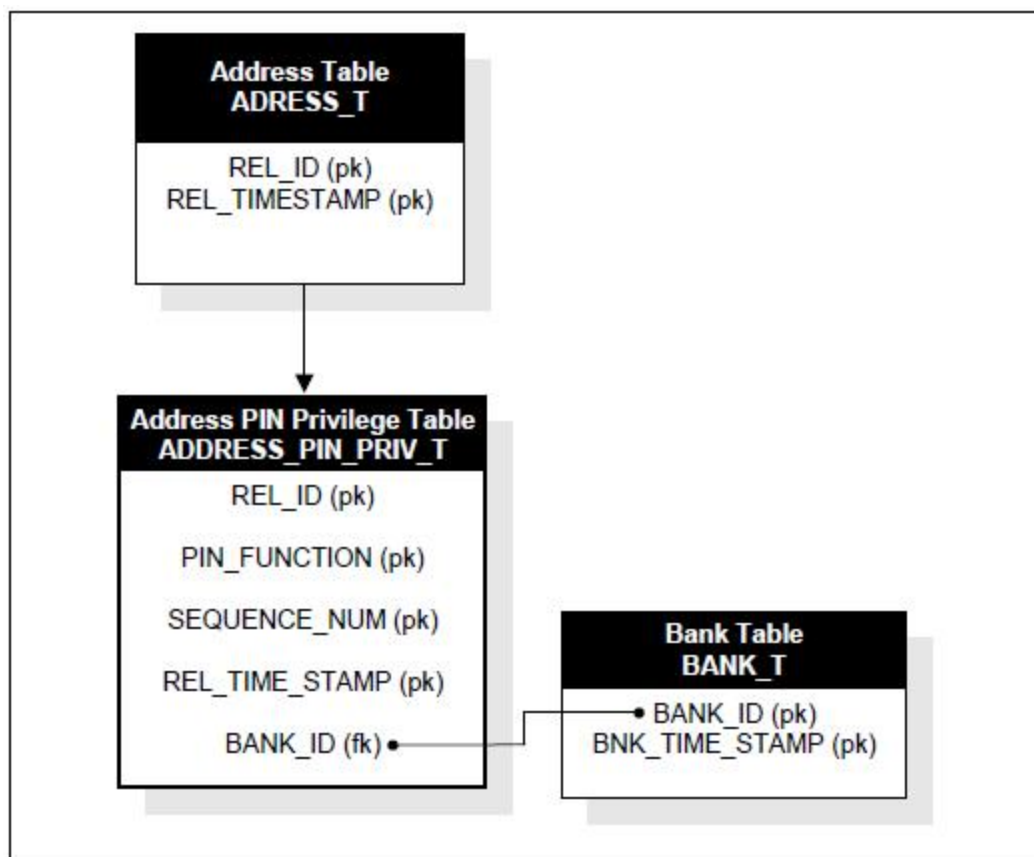
The Address PIN Privilege table sets the default dollar values for each PIN.

Table Relationships

The BANK_ID column in the Address PIN Privilege table relates to the BANK_ID column in the Bank table.

Relationship Diagram

This relationship diagram shows the Address PIN Privilege table along with its parent table, the Address table, and the Bank table.



Column Descriptions

REL_ID (pk)	(number 10). Unique reference number for this address.
-------------	--

Relationship File Tables

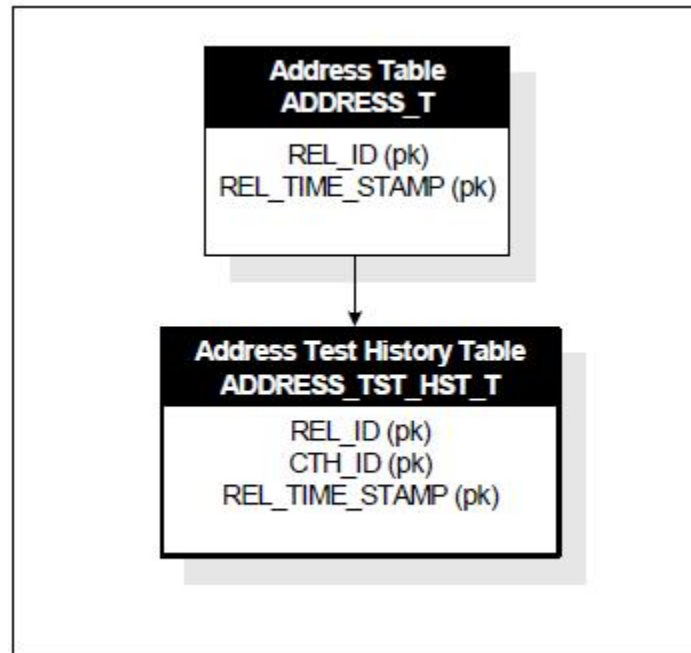
PIN_FUNCTION (pk)	(varchar 6). Function or privilege controlled by the PIN; values are configurable.
SEQUENCE_NUM (pk)	(number 10). System-generated sequential number that uniquely identifies this PIN.
REL_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
BANK_ID (fk)	(varchar 3). Bank ID of the bank to which the address belongs.
ADDRESS_ID	(varchar 5). Address identifier appended to the account number used to distinguish among multiple addresses sharing the same account ID.
PIN_TRNSF_LIM	(number 21.3). Maximum dollar value of transfers permitted with this PIN.
PIN_TEST_TYPE	(varchar 1). Type of PIN validation test; contains the following values: 1 – 8 (ID number of the bank test defined in INTRTL). T (Use telex testword authentication).
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
PIN_TRNSF_LIM_CURRENCY	(varchar 3). Transfer limit currency.

Address Test History Table (ADDRESS_TST_HST_T)

The Address Test History table provides information about telex test instructions.

Relationship Diagram

This relationship diagram shows the Address Test History table along with its parent table, the Address table and the tables related to it through foreign keys.



Column Descriptions

REL_ID (pk)	(number 10). Unique system-assigned reference number for this address.
CTH_ID (pk)	(number 10). Test history identification key for a table slot.
DIRECTION (pk)	(varchar 9). Column indicating whether message is incoming or outgoing; contains the following values: INPUT_IS or OUTPUT_IS.
REL_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
MCD	(date). Message test date.
MPDT	(date). Date and time when the message was placed.
REC_ID	(number 10). Test history record identification number.
CUR	(varchar 3). Currency of the message; used in test algorithms.
SLOT	(number 5). Slot number of the test algorithm.
SER	(number 5). Value from the test algorithm serial table.

Relationship File Tables

STS	(varchar 1). Flag indicating the transaction status; contains the following values: G (Gather) P (Placed) B (Placed bad) F (Filled) E (Empty)
TST	(varchar 16). Testword used for telex authentication.
AMT	(number 21.3). Amount of the telex transaction in base currency.
PERSON	(varchar 10). Operator ID and bank ID of the operator's identification number.
TRN_DATE	(date). Processing date portion of the transaction number indicating the date when the message was created; from the TRN (Transaction Reference Number).
TRN_NUM	(number 8). Serial number portion of the transaction number. Unique serial number assigned to the message by REFSEV (a background process in MTS that assigns the second half of the next consecutive transaction number). The first transaction message of each processing day is 00000001.
COMM_EXISTS	(varchar 1). Flag indicating whether comments have been entered (Y). or not (N).
COMM_COMMENT	(varchar 71). Comments written by the operator about the transaction's test history.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
TST_SYS_REF_NUM	(varchar 12). Test system reference number.

Message Tables

This section includes a brief description and complete list of the columns for all message tables, including funds transfer messages and administrative message templates.

- Funds Transfer Message Tables
- Message Processing Rules Tables
- Repetitive Tables
- Administrative Message Template Tables

Funds Transfer Message Tables

The Funds Transfer Message tables contain the message fields as well as advice instructions, confirmation instructions, accounting instructions, delivery instructions, and destination and history information.

Parent/child relationships

The Message table (MESSAGE_T), Message Credit table (MESSAGE_CR_T), and Message Debit table (MESSAGE_DR_T) comprise one large message record. This large record is split into three tables because of Oracle and DB2 column number limits. The three tables together function as the parent table to the following child tables:

- Message Accounting (MESSAGE_ACCTG_T)
- Message Destination (MESSAGE_DEST_T)
- Message History (MESSAGE_HIST_T)
- Message Queue (MESSAGE_QUEUE_T)
- Message Text (MESSAGE_TEXT_T)
- Message REL Amount (MESSAGE_REL_AMT_T)
- Message End of Day (MESSAGE_EOD_T)

Primary keys

All the message tables have the following primary keys in common:

- TRN_DATE
- TRN_NUMBER
- TRN_TIMESTAMP

Message Table (MESSAGE_T)

The Message table provides the fundamental message information.

Table Relationships

The BANK column in the Message table relates to the BANK_ID column in the Bank table.

The CURRENCY_CODE column in the Message table relates to the CURRENCY_ID column in the Currency table.

The BANK_ID, ID_TYPE, and ACCOUNT columns in the Account table might relate to the following columns in the Message table to provide additional information:

- Cross-bank debit account information

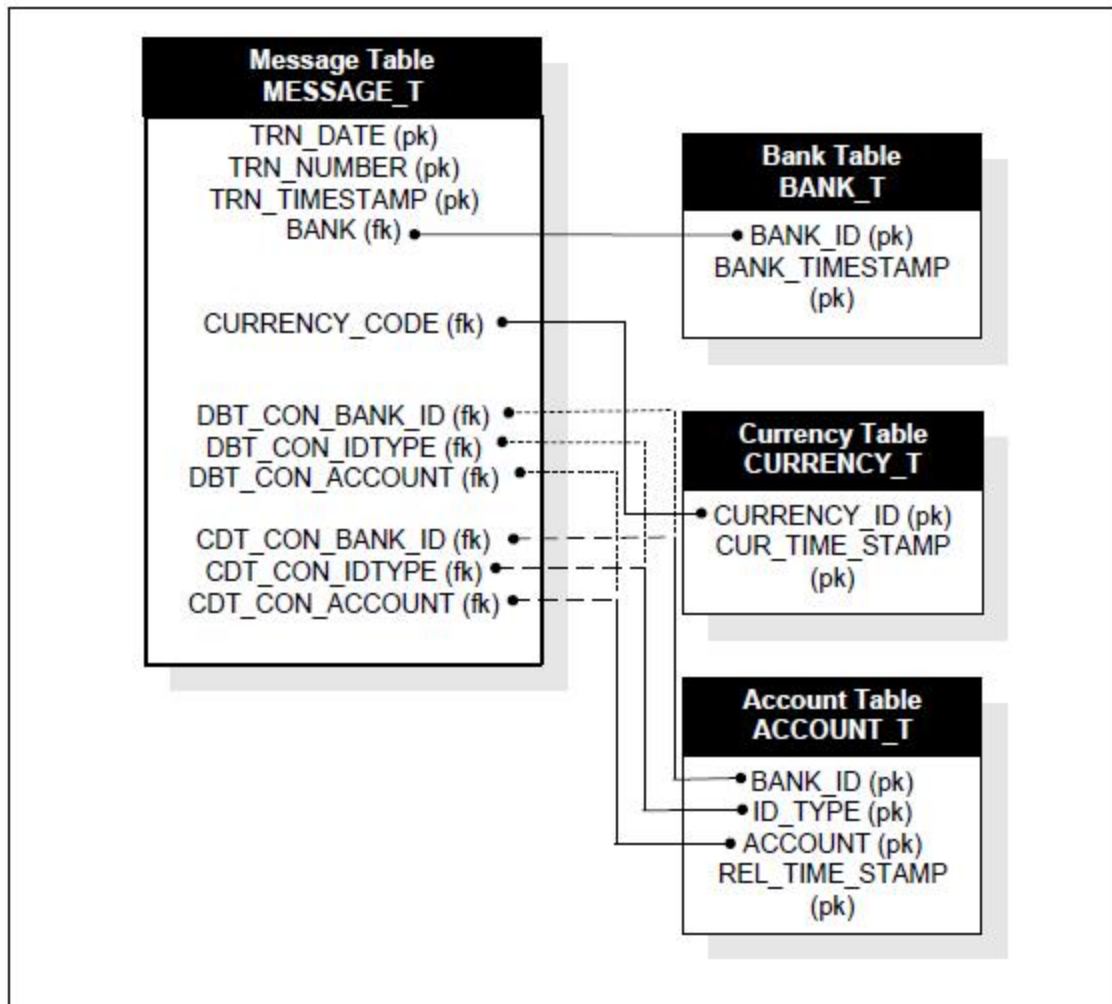
DBT_CON_BANK_ID	DBT_CON_IDTYPE	DBT_CON_ACCOUNT
-----------------	----------------	-----------------

- Cross-bank credit account information

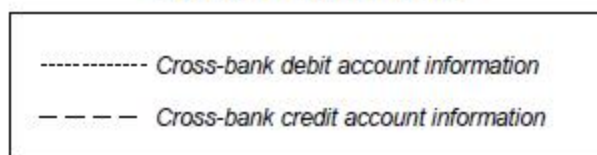
CDT_CON_BANK_ID CDT_CON_IDTYPE CDT_CON_ACCOUNT

Relationship Diagram

This relationship diagram shows the Message table along with any tables related to it through foreign keys.



Account Table Legend



Column Descriptions

TRN_DATE (pk)	(date). Date from the TRN (Transaction Reference Number) field; processing date when the message was created.
TRN_NUMBER (pk)	(number 8). Unique serial number assigned to the message by REFSERV (a background process in MTS that assigns the second half of the next consecutive transaction number). The first transaction message of each processing day is 00000001.
TRN_TIMESTAMP (pk)	(number 23). Date and time when MTS wrote this Message File Extract record; correlates to the NOW_TIMESTAMP field of the Message Extract File (MFE).
PROC_DATE	(date). Text version of the PERIOD field in yymmdd format; current processing date or date the transaction reference number (TRN) was created. This date appears the same in each record of a Message File Extract.
BANK (fk)	(varchar 3). Owning bank ID.
LOC	(varchar 6). Site-defined location (within bank) where this message was created. MTRANS is usually the standard value. No location is associated with incoming network transactions.
PAY_DATE	(date). Processing date when message accounting was performed.
PAY_TIME	(varchar 11). Processing time when message accounting was performed.
INST_DATE	(date). Instruction received date, set by input mapper of the communication link.
TRAN_TYPE	(varchar 3). Transaction type; mapped from the TYP field on the Funds Transfer (FTR). screen. This column contains the following values: ANT (Anticipation) DEP (Deposit) DFA (Draft advice) DFT (Draft) DRR (Drawdown request) DRW (Drawdown response) FFB (Fed Funds bought) FFR (Fed Funds returned) FFS (Fed Funds sold) FTR (Funds transfer) NON (Nonaccounting) PRE (Preadvised) SPL (Split accounting)

Message Tables

REPETITIVE_ID	(varchar 48). Unique identifier assigned to the repetitive template MTS uses to generate a recurring outgoing transaction. The Repetitive ID is a combination of multiple columns. The number is mapped from the RPT# field on the Funds Transfer screen. The debit party is mapped from the BNK_ID, DBT_IDTYPE, and DBT_ID fields on the Funds Transfer screen or from the customer ID of the PIN.
SOURCE_CD	(varchar 3). Transaction Source Code; mapped from the SRC field on the Funds Transfer (FTR) screen. All sources are defined in configuration files. This column contains the following values: ADJ (Adjustment) ADM (Administrative message entry function) CHP (CHIPS) CMS (Cash management system)
	DFM (Due from monitor) DOC (Letter of credit) ENT (Payment entry function) ETW (Enhanced Trading Workstation) FED (Fedwire) INT (Internal transfer) LTC (Letter requiring callback) LTR (Letter) MEM (Memo past function) MIS (Miscellaneous, for example: cash management system or remote batch entry) MTW (Money Transfer Workstation) OFL (Offline) PHN (Phone transfer initiated outside bank) RBE (Remote batch entry) SOD (Start-of-day balance load) STO (Standing order function) SWF (S.W.I.F.T.) WIR (Telex) This column also applies to any user-defined source.
INSTR_ADV_TYPE	(varchar 3). Advice method indicated by the sender for the credit party; mapped from the ADV field on the Funds Transfer (FTR). screen. This column contains the following values: CHK (Check) CHP (CHIPS) FED (Fedwire) LTR (No primary advice; secondary confirmation only) PHH (Phone hold) PHN (Phone) SWF (S.W.I.F.T.)

	<p>TKT (Online ticket)</p> <p>WIR (Telex)</p> <p>Any user-defined INTRTL advice</p>
BASE_AMOUNT	(number 21.3). Transaction amount in base currency from AMT on the Funds Transfer (FTR) screen.
TYPE_CD	(varchar 2). Type code mapped from the MTP field on the Funds Transfer (FTR) screen. Outgoing Federal Reserve, S.W.I.F.T., or CHIPS message type; correlates to the MSG_TYPE_TYP column in the Message Extract File (MFE).
SUBTYPE	(varchar 2). Subtype code mapped from the MTP field on the Funds Transfer (FTR) screen. Outgoing Federal Reserve, S.W.I.F.T., or CHIPS message subtype code; correlates to the MSG_TYPE_SUBTYP field in the Message Extract File (MFE).
IN_TYPE_CD	(varchar 2). Incoming Federal Reserve, S.W.I.F.T., or CHIPS message type code; correlates to the INMSG_TYPE_TYP field in the Message Extract File (MFE).
IN_SUBTYPE	(varchar 2). Incoming Federal Reserve, S.W.I.F.T., or CHIPS message subtype code; correlates to the INMSG_TYPE_SUBTYP field in the Message Extract File (MFE).
DBT_CHRG	(varchar 1). Flag indicating whether charges apply for debit; mapped from the CHG:DB field on the Funds Transfer (FTR) screen. Values are bank-defined, determined from configuration file routines for fee charges.
CDT_CHRG	(varchar 1). Flag indicating whether charges apply for credit; mapped from the CHG:CD field on the Funds Transfer (FTR) screen. Values are bank-defined, determined from configuration file routines for fee charges.
COMMISSION	(varchar 1). Flag indicating whether charges apply for commission; mapped from the CHG:COM field on the Funds Transfer (FTR) screen. Values are bank-defined, determined from configuration file routines for commission charges.
CBL_CHARGE	(varchar 1). Flag indicating whether charges apply for cable; mapped from the CHG:CAB field on the Funds Transfer (FTR) screen. Values are bank-defined, determined from configuration file routines for cable charges.
WIRE_TYPE	<p>(varchar 3). Wire type; used in end-of-day reporting. Site-specific combination of SRC and ADV_TYPE. The following common values, determined by configuration files, are:</p> <p>FWI (Incoming Fed wire)</p> <p>FWO (Outgoing Fed wire)</p> <p>SWI (Incoming S.W.I.F.T. wire)</p> <p>SWO (Outgoing S.W.I.F.T. wire)</p> <p>TWI (Incoming Telex wire)</p> <p>TWO (Outgoing Telex wire)</p> <p>IBT (Interbank transfer)</p> <p>blank (Default)</p>

Message Tables

STRAIGHT_THRU	<p>(varchar 1). Flag indicating whether the message was processed straight through. The flag may be set to any value the bank has in its route.cfg fully parse column. The following common values, determined by configuration files, are:</p> <p>Y: all edits passed, and the unmodified account number in the credit party of a payment or the debit party of a drawdown matched an entry in the REL file.</p> <p>N: not all of the message could be parsed; or an edit failed.</p>
	<p>A: based either on the message type or on the discovery of either a first- or second-level "autoparse" string in the message, the account number was inserted automatically.</p> <p>1: successful credit party lookup by exact account number via HOSTGET call.</p> <p>2: an account number with a valid check digit was found embedded in the name field of the credit party instead of being in the properly structured account number field. The account name found in REL or via HOSTGET matched the name in the message after normalization (if not, the straight-through level is set to 5).</p> <p>5: the credit lookup in REL or via HOSTGET succeeded but either the HOSTGET call or the check-digit edit modified the account number, for example, by adding leading zeros or inserting a branch code. Further, the normalized name returned by the lookup does not match the normalized name of the credit party (in a payment) or the debit party (in a drawdown). A memo alerts the operator to check the name.</p> <p>6: the credit party was identified with a name and no account number; there is at least one additional party beyond the credit party, which means the ultimate beneficiary is not being looked up by name only.</p> <p>7: the credit party was identified with a name and no account number. There are no further parties; the credit party is the ultimate beneficiary. If the credit party is not on the REL file, and was not found via HOSTGET, but the account number check digit is valid, then messages will autoparse, based on the FEDIN_NOF_PARSE). _FLAG value set in the bank's configuration.</p>
NETWORK_SND_IDTYPE	(varchar 1). ID type of network sender.
NETWORK_SND_ACC	(varchar 30). Account ID of network sender.
STS_ACC	(varchar 1). Flag indicating whether accounting has been performed on the message (Y). or not (N).
STS_CAN	(varchar 1). Flag indicating whether the message has been canceled (Y). or not (N).
STS_RISK	(varchar 1). Flag indicating whether the message was processed through risk (Y) or not (N).
STS_AUTO	(varchar 1). Flag indicating whether the mapper parsed the message automatically (Y) or not (N).
STS_OVER	<p>(varchar 1). The type of overdraft condition the message caused. This column contains the following values:</p> <p>N (None)</p> <p>L (Overline)</p> <p>D (Overdraft)</p>

STS_CSTMT	(varchar 1). Number of credit statement copies.
STS_DSTMT	(varchar 1). Number of debit statement copies.
CALL_LIM	(varchar 1). Indicates whether the operator overrode the caller limit.
CALL_RPT	(varchar 1). Indicates whether the caller is permitted to use repetitives.
CALL_NON_RPT	(varchar 1). Indicates whether the caller is permitted to use nonrepetitives.
CALLBACK	(varchar 1). Indicates whether the caller overrode the message callback status.
FAIL_TST	(varchar 1). Indicates whether the operator overrode a failed test.
RPTV_LIM	(varchar 1). Indicates whether a credit limit exception exists.
RPTV_CHNG	(varchar 1). Indicates whether a repetitive has changed.
CALL_NOF	(varchar 1). Indicates whether a caller can send fund transfers to not-on-file credit parties.
MON_INTERCEPT	(varchar 1). Flag set by RISK_AND_ROUTING if an intercept monitor point is matched.
RCV_DATE	(date). Calendar date when the message was created.
RCV_TIME	(varchar 11). Time when the message was created.
DLV_DATE	(date). Calendar date when the message was sent or printed.
DLV_TIME	(varchar 11). Time when the message was sent or printed.
CALLER	(varchar 35). Name of the caller who authorized the transfer.
DOC_NUM	(varchar 7). Document number; optional column used by banks in batch processing.
ITEM_NUM	(varchar 2). Item number; optional column used by banks in batch processing.
CAN_MEMO	(varchar 80). Text entered by the operator who canceled the message.
CAN_PERSON (fk)	(varchar 10). Operator ID and bank ID of the operator who canceled the message.
MEMO	(varchar 80). Informational information for the message.
PERSON	(varchar 10). Operator ID and bank ID of the operator who processed the message.
CAL	(varchar 4). Caller's internal telephone extension.
EXCH_RATE_FLG	(varchar 1). Flag that determines how the exchange rate of the message is calculated and assigned. This column contains the following values:

Message Tables

	Blank or space (Direct) I (Indirect) V (Exchange rate should be assigned on value date)
FX_TOLERANCE	(varchar 1). Flag that determines the exchange rate tolerance. This column contains the following values: Blank or space (Within tolerance) Y (Exceeds tolerance) O (Exceeds tolerance condition was overridden)
EXCH_RATE	(number 29.11). Exchange rate.
AMOUNT	(number 21.3). Transaction amount.
CURRENCY_CODE	(varchar 3). Currency code of the transaction.
TRADER_CTRL	(varchar 10). Trader control number.
FUNDS_TYPE	(varchar 1). Determines the funds type; contains the following values: N (next day funds) or S (same day funds).
FED_IMAD	(varchar 30). Federal Reserve Input Message Accountability Data.
FED_OMAD	(varchar 30). Federal Reserve Output Message Accountability Data.
FED_ISN	(varchar 6). Federal Reserve input sequence number.
FED_OSN	(varchar 6). Federal Reserve output sequence number.
SWF_ISN	(varchar 6). S.W.I.F.T. input sequence number.
SWF_OSN	(varchar 6). S.W.I.F.T. output sequence number.
CHP_ISN	(varchar 6). CHIPS input sequence number.
CHP_OSN	(varchar 6). CHIPS output sequence number.
CHP_SSN_1	(varchar 1). First character of CHIPS system sequence number.
CHP_SSN_6	(varchar 6). Old six-character CHIPS system sequence number.
SWF_IN_MIR	(varchar 28). S.W.I.F.T. input message input reference number.
SWF_OUT_MIR	(varchar 28). S.W.I.F.T. input message output reference number.
ENTRY_PERSON	(varchar 10). Operator ID and bank ID of the operator who entered the message.
VERIFY_PERSON	(varchar 10). Operator ID and bank ID of the operator who verified the message.
REPAIR_PERSON	(varchar 10). Operator ID and bank ID of the operator who repaired the message.

EXCEPT_PERSON	(varchar 10). Operator ID and bank ID of the operator who performed the exception processing.
TOT_AMT	(number 21.3). Total amount of the transaction in base currency; used only in foreign currency transactions.
CUSIP	(varchar 9). CUSIP ID for securities transactions.
CUSIP_DESC	(varchar 20). Security description for securities transactions.
CHECK_SEQ	(varchar 10). Check sequence number.
NO_PHNADV_FLG	(varchar 1). Flag indicating whether the NO NEED TO PHONE command was used in the PHN function (Y) or not (N).
EXC_MEMO	(varchar 80). Exceptions memo explaining why the operator sent the message to exceptions.
PAY_STATE	(varchar 1). Accounting pay state flag indicating which type of accounting has been performed. This column contains the following values: Blank (No accounting performed) P (Normal accounting performed) D (Debit side of split accounting performed) C (Credit side of split accounting performed) H (Debit normal, credit to holdover account) R (Debit to holdover account, credit normal)
NOTIFY	(varchar 1). Originator notification flag.
STOP_INTERCEPT	(varchar 1). Flag set by STOP_CHECK if an intercept stop descriptor is matched.
RELEASE_TIME_HH	(varchar 2). Hour portion of the 24-hour time when the message was to be released; contains values 00 through 23, or spaces.
RELEASE_TIME_MM	(varchar 2). Minutes portion of the time when the message was to be released; contains values 00 through 59, or spaces.
PRE_FOUND	(varchar 1). Flag indicating whether PRE standing instruction is found (Y) or not (N).
FX_FOUND	(varchar 1). Flag indicating whether a codeword of FX exists both in a message and in a configuration file.
COR_DONE	(varchar 1). Reserved for custom use.
RTE_DONE	(varchar 1). Flag indicating whether RTE (route transaction to exception) standing instruction was executed and accepted. This column contains the following values: Y (Executed and accepted) N (Executed and rejected)

Message Tables

	P (Pending)
PIN_LAST_NAME	(varchar 15). Last name of caller whose PIN was used.
PIN_FIRST_NAME	(varchar 15). First name of caller whose PIN was used.
OVER_DRAFT_AMT	(number 21.3). Overdraft amount.
DBT_CON_BANK_ID (fk)	(varchar 3). Bank ID of debit concentration account.
DBT_CON_ID_TYPE (fk)	(varchar 1). Debit concentration account ID type. This column contains the following values: D (DDA) V (Savings account number)
DBT_CON_ACCOUNT (fk)	(varchar 30). Debit concentration account number.
CDT_CON_BANK_ID (fk)	(varchar 3). Bank ID of credit concentration account.
CDT_CON_ID_TYPE (fk)	(varchar 1). Credit concentration account ID type. This column contains the following values: D (DDA) V (Savings account number)
CDT_CON_ACCOUNT (fk)	(varchar 30). Credit concentration account number.
DBT_SYS_OF_REC	(varchar 3). Debit system of record flag. If debit authorization is from another system, this column has a value.
CDT_SYS_OF_REC	(varchar 3). Credit system of record flag. If credit authorization is from another system, this column has a value.
DBT_PARENT_CODE	(varchar 10). Debit account parent code. A parent code groups accounts for special processing (such as end-of-day processing in MFE). MTS does not process this column.
CDT_PARENT_CODE	(varchar 10). Credit account parent code. A parent code is a method of grouping accounts for special processing (such as end-of-day processing in MFE). MTS does not process this column.
INST_TIME	(varchar 11). Time the instruction was received.
RECEIPT_TIME	(varchar 11). Date and time when the message was received.
FRONTEND_REF	(varchar 16). Front end system reference number.
STS_CADV	(varchar 1). Credit advice status.
STS_DADV	(varchar 1). Debit advice status.
AVAIL_BAL	(number 21.3). Available balance.

LEDGER_BAL	(number 21.3). Ledger balance.
OVER_TIMESTAMP	(date). Overdraft timestamp.
RECORD_EXPIRED	(varchar 23). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
TRADE_PARTY_SUSP_FLG	(varchar 1). Indicates whether the trade party was suspended.
OTHER_PARTY_SUSP_FLG	(varchar 1). Indicates whether the other party was suspended.
LAST_USED_DATE	(date). Date on which the repetitive was last used.
ORIG_INST_VAL_DATE	(date). Original instructed value date.
TRADE_TYPE	(varchar 8). BFX trade type.
TRADE_SUBTYPE	(varchar 6). BFX trade subtype.
TRADE_BUY_SELL_FLG	(varchar 1). Indicates either: B = Buy S = Sell
TRADE_UPDATE_PENDING_FLG	(varchar 1). Indicates whether to rescind or amend pending on the trade.
ORIG_EXCH_RATE	(number 29.11). Original exchange rate.
RECEIVE_CHAR_COUNT	(varchar 11). Number of characters in the original inbound Telex or S.W.I.F.T. message, from the start of the header to the end of the text. MTS counts the number of uncompressed characters for the who Telex and S.W.I.F.T messages and message types and stores the total.
INCOMING_FORMAT	(varchar 4). Incoming message format.
INCOMING_REF	(varchar 80). External input reference.
FUNDS_CATEGORY	(varchar 10). Funds category from the batch message reference.
SEGMENT_INDEX	(number 11). Index number if there is a split from a segmented TELEX.
SPLIT_CTR	(varchar 11). Indicates that MT102 child transactions were created.
RPR_LEVEL	(varchar 2). Indicates the level of parsing in the incoming mapper.
POST_DATE	(date). Debit posting date.
POST_TIME	(varchar 11). Debit posting time.

Message Tables

SEND_DATE	(date). Send date.
FEDFUND_TYPE	(varchar 2). Fedfund type.
FED_DAYS	(varchar 11). Number of days for a fed funds message.
INTEREST_RATE	(number 7.5). Interest rate for a fed funds message.
ANT_TOL_AMT	(number 21.3). Tolerance amount for an anticipation matching.
CAL_OVR	(varchar 1). Caller override flag. Set to * during entry when the sending bank's branch and the debit party are left blank. * indicates there was no caller.
PIN_REF_ID	(varchar 11). PIN reference number.
VFC_PIN_REF_ID	(varchar 11). PIN used in the previous verify callback.
PHN_NOF_ID	(varchar 24). This matches the PIN_REF_INDEX customer ID.
TESTKEY_IN	(varchar 16). Testkey in.
TEST_TEXT	(varchar 24). Used to compute the test key.
RISKINESS_IND	(varchar 2). Riskiness indicator.
COMM_MODE	(varchar 1). Commission charge from either the debit or credit address.
CBL_MODE	(varchar 1). Cable charge from either the debit or credit address.
ADM_LOCK_TEXT_FLG	(varchar 1). Set when in fill-in-the-blanks mode.
BNK_TX_FLG	(varchar 1). Bank TX flag.
MULT_DBT_FLG	(varchar 1). Set when there are multiple debit parties.
MULT_CDT_FLG	(varchar 1). Set when there are multiple credit parties.
AMT_VFY_FLG	(varchar 1). Set when the key verify is required.
CUR_VFY_FLG	(varchar 1). Set when the key verify is required.
EXCH_VFY_FLG	(varchar 1). Set when the key verify is required.
SERIAL_VFY_FLG	(varchar 1). Set when the key verify is required.
PDM_FLG	(varchar 1). Set when any part of the message is being redelivered on an output line.
ADV_OVR_FLG	(varchar 1). Set by the input mapper if overriding the default advising.

TEST_REQ_FLG	(varchar 1). Test type, if required.
SITE_OPTION1_FLG	(varchar 1). Site specific routing flag.
ACCT_VFC_FLG	(varchar 1). Account callback limit exception (either rep or non-rep).
CDT_ADV_FLG	(varchar 1). Set by data entry when the operator overrides default data.
MULTI_CUR_FLG	(varchar 1). Multi-currency flag.
PRIORITY_FLG	(varchar 1). Current priority flag.
DELIVERY_FLG	(varchar 1). Delivery notification flag.
RTP_INTERCEPT_FLG	(varchar 1). Set by message routing if a release time payment has been held (in risk).
DELAY_FLG	(varchar 1). Indicates whether the message should not be released early.
MATCHED_ANT_FLG	(varchar 1). Matched anticipation flag.
REPAIR_CHNG_FLG	(varchar 1). Set at repair if any credit or debit data is changed.
CHIPS_RECVRY_FLG	(varchar 1). Indicates whether Chips recovery is in progress.
NON_ACCTING_FLG	(varchar 1). Indicates whether the message is a non-accounting memo post.
SWF_STMT_FLG	(varchar 1). Indicates whether the message should be included in SWIFT statements.
CHARGE_STATE_FLG	(varchar 1). Indicates whether the message is charged as straight-through.
VFY_COUNT_FLG	(varchar 1). Number of times messages have gone through VFY, not counting HIV.
MPA_DBT_AUTH_FLG	(varchar 1). Indicates whether the debit is authorized by an external DDA system. Y = Authorization was done R = Authorization must be reversed Space = Must decide what needs to be done N = Authorization does not need to be done X/space = Once reversed, a payment may have space in the flag or it may have an X in one or more of the flags due to MPA noting that would need to be done in order to process the item.
MPA_DBT_POST_FLG	(varchar 1). Indicates whether the debit posted to an external DDA system. Y = Posting was done R = Posting must be reversed Space = Must decide what needs to be done N = Posting does not need to be done

Message Tables

	X/space = Once reversed, a payment may have space in the flag or it may have an X in one or more of the flags due to MPA noting that would need to be done in order to process the item.
MPA_CDT_AUTH_FLG	(varchar 1). Indicates whether the credit is authorized by an external DDA system. Y = Authorization was done R = Authorization must be reversed Space = Must decide what needs to be done N = Authorization does not need to be done X/space = Once reversed, a payment may have space in the flag or it may have an X in one or more of the flags due to MPA noting that would need to be done in order to process the item.
MPA_CDT_POST_FLG	(varchar 1). Indicates whether the debit posted to an external DDA system. Y = Posting was done R = Posting must be reversed Space = Must decide what needs to be done N = Posting does not need to be done X/space = Once reversed, a payment may have space in the flag or it may have an X in one or more of the flags due to MPA noting that would need to be done in order to process the item.
TEST_STATE_FLG	(varchar 1). Test or authorization result.
VFC_COUNT_FLG	(varchar 1). Number of times the message has gone through VFC.
FUNDED_MSG_FLG	(varchar 1). Indicates whether the message (MPSA). is found.
PMT_CHRG_FLG	(varchar 1). Payment charge flag.
INT_REROUTE_FLG	(varchar 1). Indicates whether the message is internally re-routed.
CUR_HOLD_FLG	(varchar 1). Indicates whether the hold flag for the currency is set.
OVR_OUT_LIM_FLG	(varchar 1). Indicates whether additional currency rate review is required.
FORCE_POST_FLG	(varchar 1). Indicates whether the message has been force posted.
RPTD_SWF_STMT_FLG	(varchar 1). Indicates whether the transaction was already reported on the SWIF statement.
CHECK_FOUND	(varchar 1). Indicates whether the check standing instruction was found.
MATCH_DBT_ID	(varchar 1). Anticipation match debit ID.
MATCH_CDT_ID	(varchar 1). Anticipation match credit ID.
MATCH_BNP_ID	(varchar 1). Anticipation match beneficiary ID.

MATCH_SRN	(varchar 1). Anticipation match sender reference number.
MATCH_SIDE	(varchar 1). Indicates which side of the anticipation to match.
UMBRELLA_FLG	(varchar 1). Indicates whether automated partial matching is allowed for anticipations.
BANK_OPERATION_CODE	(varchar 4). Contents of field 23B from MT103 message.
CHG_ACC	(varchar 1). Summary field for the set of account numbers associated with the debit, credit, and beneficiary parties.
CHG_BIC	(varchar 1). Change flag for any bank identifier code (BIC). that is changed in any party.
CHG_DBT	(varchar 1). Change flag for the debit party (DBT).
CHG_SBK	(varchar 1). Change flag for the sending bank (SBk)
CHG_OBK	(varchar 1). Change flag for the origination bank (OBk)
CHG_ORP	(varchar 1). Change flag for the origination party (ORP).
CHG_INS	(varchar 1). Change flag for the instructing bank (INS).
CHG_CDT	(varchar 1). Change flag for the credit party (CDT).
CHG_IB1	(varchar 1). Change flag for the first intermediary bank (IB1).
CHG_IBK	(varchar 1). Change flag for the intermediary bank (IBk).
CHG_BBK	(varchar 1). Change flag for the beneficiary bank (BBk).
CHG_BNP	(varchar 1). Change flag for the beneficiary party (BNP).
CHG_BBI	(varchar 1). Change flag for bank-to-bank information (BBI).
CHG_OBI	(varchar 1). Change flag for ordering party-to-beneficiary information (OBI).
CHG_DTE	(varchar 1). Change flag for the change flag for all dates.
CHG_AMT	(varchar 1). Change flag for the change flag for all amounts and/or currencies and/or rate.
CHG_CNF	(varchar 1). Change flag for the confirmation instructions.
CHG_REG	(varchar 1). Change flag for the regulatory fields.
CHG_CRG	(varchar 1). Change flag for the charges information.
CHG_ADV	(varchar 1). Change flag for the credit advice type.

Message Tables

CHG_PRD	(varchar 1). Change flag for the product code.
PMT_CHAR_CODE	(varchar 10). Payment characteristics code.
CUTOFF_OVERRIDE	(varchar 1). Indicates whether the cutoff time will be ignored. Y = Cutoff time will be ignored.
EXCHANGE_RATE_STAT	(varchar 1). Rate status indicator. I = Imposed.
FOREX_BYPASS_FLAG	(varchar 1). Awaiting rate queue bypass indicator. Y = Message routing will redirect message.
GLOBAL_DEBIT	(varchar 1). Global debit indicator. Indicates whether the particular transaction is a parent, child, or individual transaction. P = Parent book transfer. C = Child transaction. Blank = Neither P nor C; normal transaction.
BOPR_PTY_SIDE_IND	(varchar 1). Balance of payment reporting (BOPR). party indicator. D = Debit. C = Credit. Blank.
FORCE_FLAG	(varchar 1). Indicates whether MPA allows an overdraft when there are no funds from Autotrieve. Y = MPA allows overdraft.
DEFINITIVE_FLAG	(varchar 1). Indicates whether dates, rates, and fees are guaranteed for the transaction. Y = dates, rates, and fees are guaranteed.
CANCELLED_FLAG	(varchar 1). Indicates the method by which the mainframe posting and authorization (MPA). process is cancelled. M = Manually cancelled. A = Automatically cancelled. Blank = Not cancelled.
FIN_CPY_SRC_FLG	(varchar 1). Set the flag to R to set the third byte of the RTGS+ 113 field to Y. Setting the flag to R means that a request is being made to create a message type 012. R = MT102 request for CDT adv.
SUPPR_PMT_CONF	(varchar 1). Indicates whether to suppress payment confirmation. Y = Suppress payment confirmation.

ENTERED_SIDE	(varchar 1). For multiple currencies, distinguishes input amounts from calculated amounts.
DUE_DATE	(date). Date on which the message is due to leave the Future Queue.
DUE_DATE_CHG	(varchar 20). Degree of change enforcement for a non-business date.
DUE_DATE_ADJ	(varchar 20). Forward or backward adjustment for the due date.
DUE_DATE_SP	(varchar 20). Indicates special circumstances under which the due date must be replaced.
PRIME_SEND_DATE	(date). Date on which the primary advice should be sent.
PRIME_SEND_DATE_CHG	(varchar 20). Degree of change enforcement when the prime sending date falls on a non-business date.
PRIME_SEND_DATE_ADJ	(varchar 20). Forward or backward adjustment for the prime sending date.
PRIME_SEND_DATE_SP	(varchar 20). Indicates special circumstances under which the send date must be replaced.
SECOND_SEND_DATE	(date). Date on which the secondary advice should be sent.
SECOND_SEND_DATE_CHG	(varchar 20). Degree of change enforcement when the secondary sending date falls on a non-business date.
SECOND_SEND_DATE_ADJ	(varchar 20). Forward or backward adjustment for the secondary sending date.
SECOND_SEND_DATE_SP	(varchar 20). Indicates special circumstances under which the second send date must be replaced.
REG_ID_TYPE	(varchar 1). Regulatory ID type.
REG_ID	(varchar 34). Regulatory ID.
BOPR_TEXT1	(varchar 35). Balance of payment reporting justification text line 1.
BOPR_TEXT2	(varchar 35). Balance of payment reporting justification text line 2.
BOPR_TEXT3	(varchar 35). Balance of payment reporting justification text line 3.
AUTH_REF	(varchar 20). Authorization reference ID from the host.
MSG_STATE	(varchar 20). Indicates the state within the processing cycle for bulk messages.
RELATED_TRN_DATE	(varchar 8). Related transaction reference number period, pointing to the parent during bulk processing.
RELATED_TRN_NUM	(varchar 8). Related transaction reference number, pointing to the parent during bulk processing.

Message Tables

CDT_POST_DATE	(date). Credit posting date.
CDT_POST_TIME	(varchar 11). Credit posting time.
OPERTNL_RLS_DATE	(date). Operational release date.
OPERTNL_RLS_TIME	(varchar 11). Operational release time.
RAMET_FLG	(varchar 1). Y = member of a clone.
NON_RET_VALVE_FLG	(varchar 1). Determines whether to allow destination operator return an item to the wire room. T = do not allow destination operator to return item to the wire room.
CUTOFF_DONE	(varchar 1). T = cutoff done, ignore subsequent.
FOREX_FLAG	(varchar 1). Y = Forex processing is completed.
FEE_FLAG	(varchar 1). Y = Fee processing completed.
EUR_BENE_CHRG_ALLOW	(varchar 1). N = Do not charge beneficiary.
HIGH_MSG_STATE	(varchar 20). The highest state reached by a message.
DBT_CHRG_ORG	(varchar 10). Origination of charges, modifiers on.
CDT_CHRG_ORG	(varchar 10). Origination of charges.
COMM_CHRG_ORG	(varchar 10). Origination of charges, Comm.
CBL_CHRG_ORG	(varchar 10). Origination of charges, CBL.
FTR_PROC_RULE_STATE	(varchar 9). FTR processing rule caching data.
POSTAL_CODE_OVR_FLG	(varchar 1). Indicates whether the postal code for the message was overridden. Y = postal code was overridden.
RPR_FLG	(varchar 1). Indicates whether the message was repaired. Y = message was repaired.
EXC_FLG	(varchar 1). Indicates whether the message went through exception processing. Y = message went through exception processing.

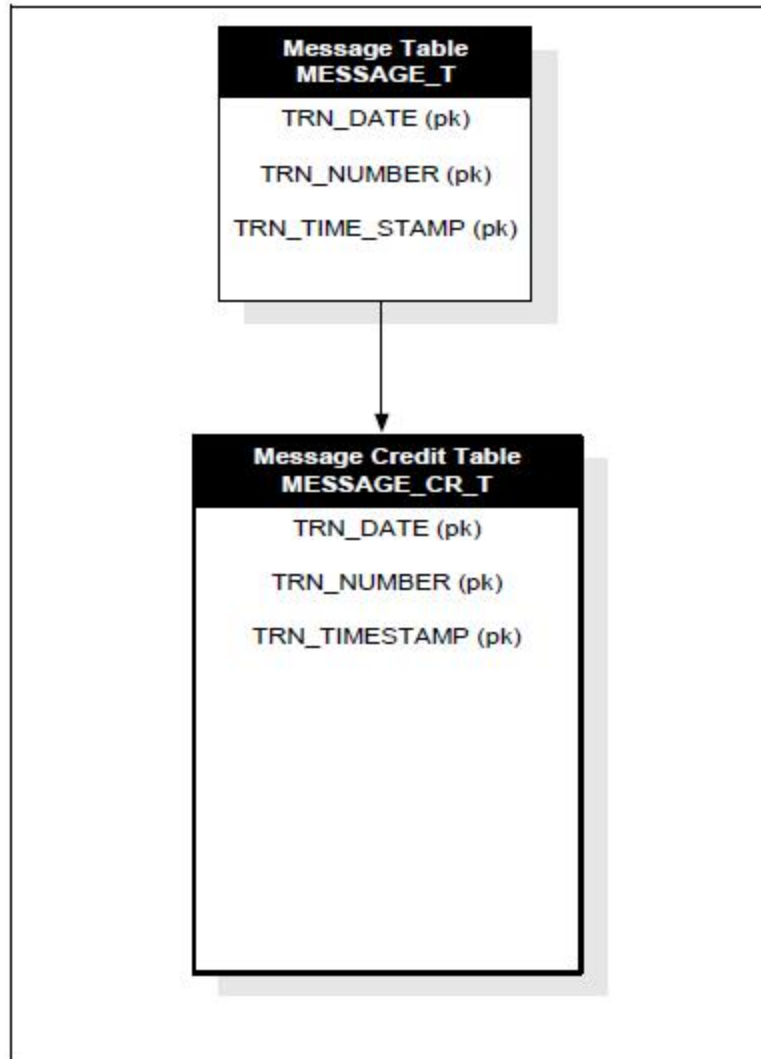
IMPOSED_AMOUNT	(varchar 1). Y = An amount was entered for determining the channel.
GLOBAL_CREDIT	(varchar 1). Global credit indicator: P C blank
PAYMNT_FUNDING_IND	(varchar 1). Indicates how the payment is funded: booked pre-funded pre_authorized DR required Authorized and DR
PAYMNT_ASYNC_FLG	(varchar 1). Payment is to be processed asynchronously.
PAYMNT_RESP_TIME	(varchar 14). ISO 8583 field 045 response time for Faster Payments.
MON_PER_INTERCEPT_FLG	(varchar 1). Indicates whether monitor points with periodic limits were either hit or overridden. H = Hit O = Overridden
MON_PER_LOG_FLG	(varchar (1). Indicates whether monitor point log records were updated. Y = One or more monitor point log records were updated.
SMPL_WRHS_INDICATOR	(varchar 1). Reserved for future use.
ENT_REF_SYS_VEC	(varchar 10). Lists the required Entry Referential Calls for the container.
WRHS_REF_SYS_VEC	(varchar 10). Status of calls for up to 10 referential systems.
REF_CALL_VECTOR	(varchar 3). Entry or Warehouse call/number.
SMPL_PYMT_FORMAT	(varchar 10). Reserved for future use.
CONTAINER_REJ_REASON	(varchar 4). Reserved for future use.
UNIQUE_REF	(varchar 16). Unique reference number for Blacklist monitoring.

Message Credit Table (MESSAGE_CR_T)

The Message Credit table provides credit information for messages.

Relationship diagram

This relationship diagram shows the Message Credit table along with its parent table, the Message table.



Column Descriptions

TRN_DATE (pk)	(date). Date from the transaction reference number (TRN) ; processing date when the message was created.
TRN_NUMBER (pk)	(varchar 8). Unique serial number assigned to the message by REFSESV (a background process in MTS that assigns the second half of the next consecutive transaction number) The first transaction message of each processing day is 00000001.

TRN_TIMESTAMP (pk)	(number 23). Date and time when this Message File Extract record was written; correlates to the NOW_TIMESTAMP column of the Message Extract File (MFE).
CDT_IDTYPE	(varchar 1). Credit party account ID type; mapped from the first part of CDT field on the Funds Transfer (FTR) screen. This column contains the following values: A (Fedwire routing ABA number) B (Branch code) C (CHIPS user ID) D (DDA) E (Extended ID) F (Foreign nostro account number) G (General ledger account number) I (Interbank lookup key) K (Customer ID number) M (Private leased line network MAC code) N (MTS mnemonic name) P (CHIPS participant ID) R (ACI Worldwide system addressreference number) S (8- or 11-character S.W.I.F.T. address) T (Telex number)
	U (User-defined name) V (Savings account number) X (Telex answerback) Z (Cable address)
CDT_ID	(varchar 64). Credit party account ID; mapped from the CDT field on the Funds Transfer (FTR) screen.
CDT_NAME1	(varchar 35). First line of the credit party account name and address.
CDT_NAME2	(varchar 35). Second line of the credit party account name and address.
CDT_NAME3	(varchar 35). Third line of the credit party account name and address.
CDT_NAME4	(varchar 35). Fourth line of the credit party account name and address.
CDT_NOF_FLAG	(varchar 1). If the credit party name is not in the REL file, but is an ABA from the AUX database, this flag is an asterisk (*). If the credit party is ambiguous, this flag is a question mark (?). Otherwise, this flag is blank.
CDT_TRAN_NAME_C1	(varchar 8). Transaction code name.
CDT_TRAN_NAME_C2	(varchar 8). Transaction code name.
CDT_TRAN_NAME_C3	(varchar 8). Transaction code name.

Message Tables

CDT_TRAN_NAME_C4	(varchar 8). Transaction code name.
CDT_TRAN_NAME_C5	(varchar 8). Transaction code name.
CDT_TRAN_NAME_C6	(varchar 8). Transaction code name.
CDT_TRAN_NAME_C7	(varchar 8). Transaction code name.
CDT_TRAN_NAME_C8	(varchar 8). Transaction code name.
CDT_TRAN_NAME_C9	(varchar 8). Transaction code name.
CDT_TRAN_NAME_C10	(varchar 8). Transaction code name.
CDT_DEPT	(varchar 20). Bank department where the credit account is assigned; mapped from the DEPT field on the Funds Transfer (FTR) screen.
CDT_SERIAL	(varchar 10). Serial portion of the test algorithm for telex messages.
CDT_ACCTG_IDTYPE	(varchar 1). Account ID type of underlying account that receives credit for this transfer; contains the following values: D (DDA) F (Foreign nostro account number) G (General ledger account number) P (CHIPS participant ID) V (Savings account number)
CDT_ACCTG_SLASH	(varchar 1). Slash separating the account type from the account number.
CDT_ACCTG_ACCOUNT	(varchar 30). Account ID of underlying account that receives credit for this transfer.
CDT_DISTRICT	(varchar 2). If the credit party is Fed and address is A/DDNNNNNNN, DD represents the credit district or is blank.
CDT_OFFICE	(varchar 10). Custom column used for special processing (such as end-of-day processing in MFE) MTS does not process this column.
CDT_ACC_CLASS	(varchar 6). Site-defined account class column; mapped from the CLASS field on the REL Account Screen.
CDT_ADR_COUNTRY	(varchar 2). Credit account's country code; mapped from the REL Account Screen. Blank defaults to US.
CDT_ADR_TYPE	(varchar 1). Credit account's relation type from the Relationship File (REL) Address Screen; contains the following values: I (Internal) B (Bank) C (Corporate)

CDT_ADR_SUBTYPE	(varchar 3). Credit account's relation subtype from the Relationship File (REL) Address Screen. Refer to the following table for possible values:
CDT_SEC_ACCTG_FLG	(number 10). Flag indicating the amount and type of account to receive secondary accounting; contains the following values: 0 (None) 1 (Same party; further debit or credit charge to accounting party) 2 (FED funds sale interest) 3 (FED funds purchase interest) 4 (Check commission clear) 5 (Dft commission clear) 6 (Commission) 7 (Cable) 8 (Other party; further charge to debit or credit party) 9 (FED funds term sale) 10 (FED funds TERM purchase interest) 11 (Charge to credit party) 12 (Informational)
CDT_SEC_ACCTG_AMT	(number 21.3). Secondary accounting amount.
CDT_TER_ACCTG_FLG	(number 10). Flag indicating the amount and type of account to receive tertiary accounting; contains the following values: 0 (None) 1 (Same party; further debit or credit charge to accounting party) 2 (Fed sale interest) 3 (Fed purchase interest) 4 (Check commission clear) 5 (Dft commission clear) 6 (Commission) 7 (Cable) 8 (Other party; further charge to debit or credit party) 9 (FED funds term sale) 10 (FED funds TERM purchase interest) 11 (Charge to credit party) 12 (Informational)
CDT_TER_ACCTG_AMT	(number 21.3). Tertiary accounting amount.
CDT_HOLD_FLG	(varchar 1). Flag set in the Relationship File (REL) The default value is N (no hold); any other value causes MTS to send the transaction to risk.

Message Tables

CDT_AMOUNT	(number 21.3). Credit dollar value of the transaction; mapped from the AMT field on the Funds Transfer (FTR) screen.
CDT_CURRENCY	(varchar 3). Credit foreign currency code; mapped from the AMT field on the Funds Transfer (FTR) screen.
CDT_SHNAM	(varchar 20). Bank short name with the Federal Reserve Bank; if this column is blank, then the first line of the name is used.
CDT_RT_CODE	(varchar 6). Bank-assigned mnemonic that provides a shortcut for routing messages to one or more queues for printing; mapped from the RTC field on the Funds Transfer (FTR) screen.
CDT_ADV_INST1	(varchar 35). First line of additional text sent to the credit party.
CDT_ADV_INST2	(varchar 35). Second line of additional text sent to the credit party.
CDT_ADV_INST3	(varchar 35). Third line of additional text sent to the credit party.
CDT_SEC_SUB_ACCT	(varchar 61). Credit or sender's sub-account (in the case of a securities transaction).
CDT_ACC_PROD_CODE	(varchar 20). Credit account product codes.
CDT_DELIVERY_FLG	(varchar 1). Credit delivery flag.
CDT_ACC_PARENT	(varchar 10). Account ID of parent account that will be credited for this transfer.
CDT_RECON_REF	(varchar 12). Account ID of reconciliation account that will be credited for this transfer. This column is also sometimes used for affiliate processing; affiliate bank ID.
CDT_CUSTOMER_ID	(varchar 20). Account ID of customer account that will be credited for this transfer.
BBK_IDTYPE	(varchar 1). ID type of the beneficiary bank account; contains the following values: Space (Foreign account) C (CHIPS user) E (Extended ID) S (S.W.I.F.T. ID) X (Telex answer-back)
BBK_ID	(varchar 64). Beneficiary bank ID. The beneficiary bank can be accessed by multiple IDs.
BBK_NAME1	(varchar 35). First line of beneficiary bank account name and address.
BBK_NAME2	(varchar 35). Second line of beneficiary bank account name and address.
BBK_NAME3	(varchar 35). Third line of beneficiary bank account name and address.

BBK_NAME4	(varchar 35). Fourth line of beneficiary bank account name and address.
BBK_ADV_INST1	(varchar 35). First line of additional text to send to the beneficiary bank; mapped from the BNF BNK ADVICE INSTRUCTIONS field on the Funds Transfer (FTR) screen.
BBK_ADV_INST2	(varchar 35). Second line of additional text to send to the beneficiary bank; mapped from the BNF BNK ADVICE INSTRUCTIONS field on the Funds Transfer (FTR) screen.
BBK_ADV_INST3	(varchar 35). Third line of additional text to send to the beneficiary bank; mapped from the BNF BNK ADVICE INSTRUCTIONS field on the Funds Transfer (FTR) screen.
BBK_SECWIRE	(varchar 1). Flag indicating whether the beneficiary bank is to be sent a secondary wire (Y) or not (N).
BNP_IDTYPE	(varchar 1). Beneficiary ID type; contains the following values: Space (Foreign account) C (CHIPS user) E (Extended ID) S (S.W.I.F.T. ID) X (Telex answer-back)
BNP_ID	(varchar 64). Beneficiary ID. The beneficiary can be accessed by multiple IDs.
BNP_NAME1	(varchar 35). First line of beneficiary account name and address.
BNP_NAME2	(varchar 35). Second line of beneficiary account name and address.
BNP_NAME3	(varchar 35). Third line of beneficiary account name and address.
BNP_NAME4	(varchar 35). Fourth line of beneficiary account name and address.
BNP_ADV_INST1	(varchar 35). First line of additional text to send to the beneficiary bank; mapped from the BNF BNK ADVICE INSTRUCTIONS field on the Funds Transfer (FTR) screen.
BNP_ADV_INST2	(varchar 35). Second line of additional text to send to the beneficiary bank; mapped from the BNF BNK ADVICE INSTRUCTIONS field on the FTR screen.
BNP_ADV_INST3	(varchar 35). Third line of additional text to send to the beneficiary bank; mapped from the BNF BNK ADVICE INSTRUCTIONS field on the FTR screen.
BNP_BNK_FLG	(varchar 1). Flag indicating whether the beneficiary party is a bank (Y) or not (N); mapped from the BK field on the FTR screen.
BNP_CHARGE_FLG	(varchar 1). Flag indicating who to charge; mapped from the CH field on the FTR screen. This column contains the following values: B (Beneficiary) or O (Originator).

Message Tables

IBK_IDTYPE	(varchar 1). Intermediary bank ID type; contains the following values: Space (Foreign account) C (CHIPS user) E (Extended ID) S (S.W.I.F.T. ID) X (Telex answer-back)
IBK_ID	(varchar 64). Intermediary bank identifier. The intermediary bank can be accessed by multiple IDs.
IBK_NAME1	(varchar 35). First line of intermediary bank name and address.
IBK_NAME2	(varchar 35). Second line of intermediary bank name and address.
IBK_NAME3	(varchar 35). Third line of intermediary bank name and address.
IBK_NAME4	(varchar 35). Fourth line of intermediary bank name and address.
IBK_ADV_INST1	(varchar 35). First line of additional text to send to the intermediary bank; mapped from the INTR BNK ADVICE INSTRUCTIONS field on the Funds Transfer (FTR) screen.
IBK_ADV_INST2	(varchar 35). Second line of additional text to send to the intermediary bank; mapped from the INTR BNK ADVICE INSTRUCTIONS field on the Funds Transfer (FTR) screen.
IBK_ADV_INST3	(varchar 35). Third line of additional text to send to the intermediary bank; mapped from the INTR BNK ADVICE INSTRUCTIONS field on the Funds Transfer (FTR) screen.
IBK_SECWIRE	(varchar 1). Flag that determines whether the intermediary bank is sent a secondary wire (Y) or not (N).
IB1_IDTYPE	(varchar 1) First intermediary bank ID type; contains the following values: Space (Foreign account) C (CHIPS user) E (Extended ID) S (S.W.I.F.T. ID) X (Telex answer-back)
IB1_IB1_ID	(varchar 64). First intermediary bank identifier. The first intermediary bank can be referenced by multiple IDs.
IB1_NAME1	(varchar 35). First line of first intermediary bank name and address.
IB1_NAME2	(varchar 35). Second line of first intermediary bank name and address.
IB1_NAME3	(varchar 35). Third line of first intermediary bank name and address.

IB1_NAME4	(varchar 35). Fourth line of first intermediary bank name and address.
IB1_ADV_INST1	(varchar 35). First line of additional text to send to the first intermediary bank; mapped from the INTR BNK ADVICE INSTRUCTIONS field on the FTR screen.
IB1_ADV_INST2	(varchar 35). Second line of additional text to send to the first intermediary bank; mapped from the INTR BNK ADVICE INSTRUCTIONS field on the FTR screen.
IB1_ADV_INST3	(varchar 35). Third line of additional text to send to the first intermediary bank; mapped from the INTR BNK ADVICE INSTRUCTIONS field on the FTR screen.
ORP_BEN_INF1	(varchar 35). First line of additional text to send to the beneficiary party from the originating party; mapped from the ORIG TO BEN INFO field on the FTR screen.
ORP_BEN_INF2	(varchar 35). Second line of additional text to send to the beneficiary party from the originating party; mapped from the ORIG TO BEN INFO field on the FTR screen.
ORP_BEN_INF3	(varchar 35). Third line of additional text to send to the beneficiary party from the originating party; mapped from the ORIG TO BEN INFO field on the FTR screen.
ORP_BEN_INF4	(varchar 35). Fourth line of additional text to be sent to the beneficiary party from the originating party; mapped from the ORIG TO BEN INFO field on the FTR screen.
RECORD_EXPIRED	(number 23). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
IB1_CHIPS_UPTO	(number 10). First intermediary bank's length of the part not from the CHIPS UID database.
IB1_CHIPS_QUAL	(number 10). First intermediary bank's length of the part from the CHIPS UID database.
IBK_CHIPS_UPTO	(number 10). Intermediary bank's length of the part not from the CHIPS UID database.
IBK_CHIPS_QUAL	(number 10). Intermediary bank's length of the part from the CHIPS UID database.
BBK_CHIPS_UPTO	(number 10). Beneficiary bank's length of the part not from the CHIPS UID database.
BBK_CHIPS_QUAL	(number 10). Beneficiary bank's length of the part from the CHIPS UID database.
BNP_CHIPS_UPTO	(number 10). Beneficiary's length of the part not from the CHIPS UID database.
BNP_CHIPS_QUAL	(number 10). Beneficiary's length of the part from the CHIPS UID database.
CDT_CURRENCY_SUSP_FLG	(varchar 1). Indicates whether the credit currency was suspended.

Message Tables

CDT_LIM_FLG	(varchar 1). Indicates whether there is a risk condition on credit transaction limits.
CDT_EXCH_RATE	(number 29.11). Calculated exchanged rate.
CDT_ADR_BNK_ID	(varchar 3). Credit party address bank ID.
CDT_REL_ID	(varchar 11). Credit party address reference ID.
CDT_ACC_COUNTRY	(varchar 2). Credit account country.
CDT_ADR_CLASS	(varchar 6). Credit address class.
CDT_ADR_CITY	(varchar 15). Credit address city.
CDT_ADV_TYP	(varchar 3). Advice method.
CDT_ADR_ADV_TYPE	(varchar 3). Advice method indicated by credit address.
CDT_LOCATION	(varchar 6). Credit location.
CDT_RT_STATE	(varchar 1). Indicates the routing code instructions are queued.
CDT_WIR_IDTYPE	(varchar 1). ID type for the credit wire key.
CDT_WIR_ID	(varchar 64). ID for the credit wire key.
PYMNT_ADV_TYPE	(varchar 3). Final payment information.
PYMNT_PHN_TIME	(varchar 4). Payment phone time.
PYMNT_Q	(varchar 65). Payment queue.
CDT_PAY_FLG	(varchar 1). State of payment instruction.
CDT_SECWIR_FLG	(varchar 1). State of secondary wire instruction.
CDT_ALT_SWIFT_FLG	(varchar 1). The CDT_WIR_KEY field contains a copy of the SI_ROUTE_ID for the primary wire.
CDT_ID_CHG_FLG	(varchar 1). Indicates whether the credit ID has changed.
CDT_NOF_LOOKED_UP	(varchar 1). Indicates whether the not-on-file credit party has been looked up.
IB1_ADR_BNK_ID	(varchar 3). First intermediary bank party address bank ID.
IB1_REL_ID	(varchar 11). First intermediary bank party address reference ID.
IB1_SECWIR	(varchar 1). Flag indicating whether the first intermediary bank is sent a secondary wire (Y) or not (N).

IBK_ADR_BNK_ID	(varchar 3). Intermediary bank party address bank ID.
IBK_REL_ID	(varchar 11). Intermediary bank party address reference ID.
BBK_ADR_BNK_ID	(varchar 3). Beneficiary bank party address bank ID.
BBK_REL_ID	(varchar 11). Beneficiary bank party address reference ID.
CDT_PAYSYS_FMT	(varchar 1). User payment system format indicator.
CDT_STATE	(varchar 3). State or province of the credit party.
CDT_POSTAL_CODE	(varchar 15). Postal code of the credit party.
CDT_RES_COUNTRY	(varchar 2). Resident country of the credit party.
CDT_REF_NUM	(varchar 16). Reference number of the credit party.
IBK_STATE	(varchar 3). State or province of the intermediary bank party.
IBK_POSTAL_CODE	(varchar 15). Postal code of the intermediary bank party.
IBK_RES_COUNTRY	(varchar 2). Resident country of the intermediary bank party.
IBK_REF_NUM	(varchar 16). Reference number of the intermediary bank party.
IB1_STATE	(varchar 3). State or province of the first intermediary bank party.
IB1_POSTAL_CODE	(varchar 15). Postal code of the first intermediary bank party.
IB1_RES_COUNTRY	(varchar 2). Resident country of the first intermediary bank party.
IB1_REF_NUM	(varchar 16). Reference number of the first intermediary bank party.
BBK_STATE	(varchar 3). State or province of the beneficiary bank party.
BBK_POSTAL_CODE	(varchar 15). Postal code of the beneficiary bank party.
BBK_RES_COUNTRY	(varchar 2). Resident country of the beneficiary bank party.
BBK_REF_NUM	(varchar 16). Reference number of the beneficiary bank party.
BNP_STATE	(varchar 3). State or province of the beneficiary party.
BNP_POSTAL_CODE	(varchar 15). Postal code of the beneficiary party.
BNP_RES_COUNTRY	(varchar 2). Resident country of the beneficiary party.
BNP_REF_NUM	(varchar 16). Reference number of the beneficiary party.

Message Tables

BNP_MAILING_COUNTRY	(varchar 2). Mailing country of the beneficiary party.
REGULATORY_REPORT1	(varchar 35). First field of the regulatory report (77B).
REGULATORY_REPORT2	(varchar 35). Second field of the regulatory report (77B).
REGULATORY_REPORT3	(varchar 35). Third field of the regulatory report (77B).
CDT_AUTH_REF	(varchar 34). Authorization reference ID returned from the host during MPA processing.
CDT_VALUE_DATE	(date). Date on which funds are presumed to be removed, for interest purposes.
CDT_VALUE_DATE_CHG	(varchar 20). Degree of change enforcement for a non-business date.
CDT_VALUE_DATE_ADJ	(varchar 20). Forward or backward adjustment for the credit value date.
CDT_VALUE_DATE_SP	(varchar 20). Indicates special circumstances under which the credit value date must be replaced.
CDT_BOOK_DATE	(date). Book date, or when debit accounting is posted to the mainframe.
CDT_BOOK_DATE_CHG	(varchar 20). Degree of change enforcement for a non-business date.
CDT_BOOK_DATE_ADJ	(varchar 20). Forward or backward adjustment for the credit book date.
CDT_BOOK_DATE_SP	(varchar 20). Indicates special circumstances under which the credit book date must be replaced.
BNP_ADR_BNK_ID	(varchar 3). Beneficiary party address bank ID.
BNP_REL_ID	(varchar 11). Beneficiary party address reference ID.
CDT_CAMEFROM	(varchar 20). Source where the credit party was found.
IB1_CAMEFROM	(varchar 20). Source where the first intermediary bank party was found.
IBK_CAMEFROM	(varchar 20). Source where the intermediary bank party was found.
BBK_CAMEFROM	(varchar 20). Source where the beneficiary bank party was found.
BNP_CAMEFROM	(varchar 20). Source where the beneficiary party was found.
CREDITSIDE_RESIDENCY	(varchar 2). Residency of the ultimate credit party, for example, the originating party, ordering bank, or credit party.
CDT_PROFILE_IDBANK	(varchar 3). Bank for the credit party profile.
CDT_PROFILE_IDTYPE	(varchar 1). ID type for the credit party profile.
CDT_PROFILE_IDACC	(varchar 30). Account number for the credit party profile.

CDT_PROFILE_IDADR	(varchar 5). Address code for the credit party profile.
CDT_PROFILE_IDPAD	(varchar 1). Padding for the credit party profile.
CDT_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the credit party.
IBK_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the intermediary bank party.
IB1_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the first intermediary bank party.
BBK_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the beneficiary bank party.
BNP_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the beneficiary party.
CDT_BILAT_IDBANK	(varchar 3). Bank ID for the credit bi-lateral account.
CDT_BILAT_IDTYPE	(varchar 1). ID type for the credit bi-lateral account.
CDT_BILAT_IDKEY	(varchar 30). Account number for the credit bi-lateral account.
CDT_SEC_IDBANK	(varchar 3). Bank ID for credit secondary account.
CDT_SEC_IDTYPE	(varchar 1). ID type for credit secondary account.
CDT_SEC_IDKEY	(varchar 30). Account number for the credit secondary account.
CDT_SEC_ACCTG_CUR	(varchar 3). Credit secondary accounting currency.
CDT_TER_IDBANK	(varchar 3). Bank ID for the credit tertiary account.
CDT_TER_IDTYPE	(varchar 1). ID type for the credit tertiary account.
CDT_TER_IDKEY	(varchar 30). Account number for the credit tertiary account.
CDT_TER_ACCTG_CUR	(varchar 3). Credit tertiary accounting currency.
CDT_SPARE_ACC	(varchar 30). Account number for the credit spare key.
CDT_SPARE_ADR	(varchar 5). Address code for the credit spare key.
CDT_SPARE_PAD	(varchar 1). Padding for the credit spare key.
CDT_PROC_RULE_STATE	(varchar 9). Credit processing rule for caching data.
CDT_SPC_INST1	(varchar 35). Credit special instructions, line 1. Special instructions are used to recreate the existing full transaction report that lists all transactions for a specific business day in a standard MTS full message print format.
CDT_SPC_INST2	(varchar 35). Credit special instructions, line 2. Special instructions are used to recreate the existing full transaction report that lists all transactions for a specific business day in a standard MTS full message print format.

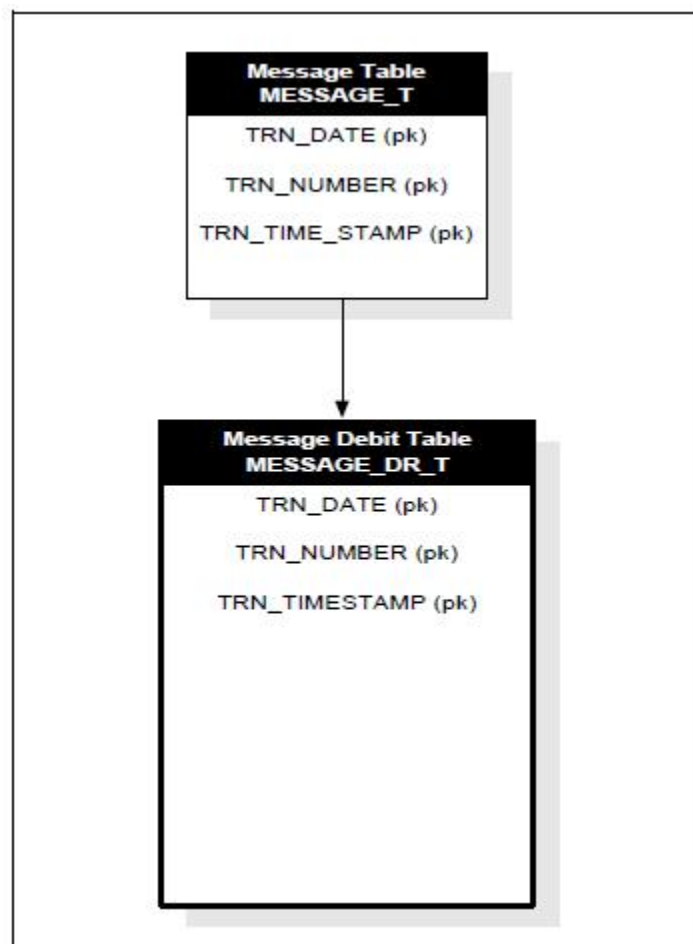
CDT_SPC_INST3	(varchar 35). Credit special instructions, line 3. Special instructions are used to recreate the existing full transaction report that lists all transactions for a specific business day in a standard MTS full message print format.
CDT_COMM_CBL_FLG	(varchar 1). C = Display REL account charges.

Message Debit Table (MESSAGE_DR_T)

The Message Debit table provides debit party information for messages.

Relationship diagram

This relationship diagram shows the Message Debit table along with its parent table, the Message table.



Column Description

TRN_DATE (pk)	(date). Date from the Transaction Reference Number (TRN); processing date on which the message was created.
TRN_NUMBER (pk)	(varchar 8). Unique serial number assigned to the message by REFSERV (a background process in MTS that assigns the second half of the next consecutive transaction number). The first transaction message of each processing day is 00000001.
TRN_TIMESTAMP (pk)	(number 23). Date and time when this Message File Extract record was written; correlates to the NOW_TIMESTAMP field of the Message Extract File (MFE).
DBT_IDTYPE	(varchar 1). Debit party account ID type; mapped from the first part of DBT field on the FTR screen. This column contains the following values: A (Fedwire routing ABA number) B (Branch code) C (CHIPS universal ID) D (DDA) E (Extended ID) F (Foreign nostro account number) G (General ledger account number) I (Interbank lookup key) K (Customer ID number) M (Private leased line network MAC code) N (MTS mnemonic name) P (CHIPS participant ID) R (ACI Worldwide system address reference number) S (S.W.I.F.T. address, 8 or 11 characters) T (Telex number) U (User-defined name)
	V (Savings account number) X (Telex answerback) Z (Cable address)
DBT_ID	(varchar 64). Debit party account ID; mapped from the DBT field on the FTR screen.
DBT_NAME1	(varchar 35). First line of the debit party account name and address.
DBT_NAME_2	(varchar 35). Second line of the debit party account name and address.
DBT_NAME_3	(varchar 35). Third line of the debit party account name and address.
DBT_NAME_4	(varchar 35). Fourth line of the debit party account name and address.

Message Tables

DBT_NOF_FLAG	(varchar 1). If the debit party name is not in the REL file, but is an ABA from the AUX database, this flag is an asterisk (*). If the debit party is ambiguous, this flag is a question mark (?). Otherwise, this flag is blank.
DBT_TRAN_NAME_C1	(varchar 8). Transaction code name.
DBT_TRAN_NAME_C2	(varchar 8). Transaction code name.
DBT_TRAN_NAME_C3	(varchar 8). Transaction code name.
DBT_TRAN_NAME_C4	(varchar 8). Transaction code name.
DBT_TRAN_NAME_C5	(varchar 8). Transaction code name.
DBT_TRAN_NAME_C6	(varchar 8). Transaction code name.
DBT_TRAN_NAME_C7	(varchar 8). Transaction code name.
DBT_TRAN_NAME_C8	(varchar 8). Transaction code name.
DBT_TRAN_NAME_C9	(varchar 8). Transaction code name.
DBT_TRAN_NAME_C10	(varchar 8). Transaction code name.
DBT_DEPT	(varchar 20). Bank department where the debit account is assigned; mapped from DEPT field on the FTR screen.
DBT_SERIAL	(varchar 10). Serial portion of the test algorithm for telex messages.
DBT_ACCTG_IDTYPE	(varchar 1). Account ID type of underlying account debited for this transfer. This column contains the following values: D (DDA) F (Foreign nostro account number) G (General ledger account number) P (CHIPS participant ID) V (Savings account number)
DBT_ACCTG_SLASH	(varchar 1). Identifies the slash between the account type and account number.
DBT_ACCTG_ACCOUNT	(varchar 30). Account ID of underlying account debited for this transfer.
DBT_DISTRICT	(varchar 2). If debit party is Fed and address is A/DDNNNNNNNN, DD represents the Federal credit district or is blank.
DBT_OFFICE	(varchar 10). Custom column used for special processing (such as end-of-day processing in MFE). MTS does not process this column.
DBT_ACC_CLASS	(varchar 6). Site-defined account class column; mapped from the CLASS field on the REL Account Screen.

DBT_ADR_COUNTRY	(varchar 2). Debit account's S.W.I.F.T. country code; mapped from the REL Account Screen. Blank defaults to US.
DBT_ADR_TYPE	(varchar 1). Debit account's relation type from the Relationship File (REL). ; contains the following values: B (Bank) C (Corporate) I (Internal)
DBT_ADR_SUBTYP	(varchar 3). Debit account's relation subtype from the Relationship File (REL). Refer to the following table for possible values:

DBT_SEC_ACCTG_FLG	(number 10). Flag indicating the amount and type of account to receive secondary accounting. This column contains the following values: 0 (None) 1 (Same party; further debit or credit charge to accounting party) 2 (FED funds sale interest) 3 (FED funds purchase interest) 4 (Check commission clear) 5 (Dft commission clear) 6 (Commission) 7 (Cable) 8 (Other party; further charge to debit or credit party) 9 (FED funds term sale) 10 (FED funds TERM purchase interest) 11 (Charge to credit party) 12 (Informational)
DBT_SEC_ACCTG_AMT	(number 21.3). Secondary accounting amount.
DBT_TER_ACCTG_FLG	(number 10). Flag indicating the amount and type of account to receive tertiary accounting. This column contains the following values: 0 (None) 1 (Same party; further debit or credit charge to accounting party) 2 (FED sale interest) 3 (FED purchase interest) 4 (Check commission clear) 5 (Debit commission clear) 6 (Commission) 7 (Cable) 8 (Other party; further charge to debit or credit party)

Message Tables

	9 (FED funds term sale) 10 (FED funds TERM purchase interest) 11 (Charge to credit party) 12 (Informational)
DBT_TER_ACCTG_AMT	(number 21.3). Tertiary accounting amount.
DBT_LIM_FLG	(varchar 1). Flag set by data entry when the operator overrides the transaction limit for the debit account.
DBT_HOLD_FLG	(varchar 1). Flag set in the Relationship File (REL). The preferred value is N (no hold); any other value causes the transaction to be sent to risk.
DBT_BAL_FLG	(varchar 1). Exception flag for the debit account balance.
CHIPS_LIM_FLG	(varchar 1). Flag set by data entry when the operator overrides the CHIPS limit for the debit account.
DBT_GRP_BAL_FLG	(varchar 1). Exception flag for the debit group balance.
DBT_PRE_FLG	(varchar 1). Debit pre-advise account limit exception. This column contains the following values: E (preadvise limit on debit) or O (override).
DBT_GRP_PRE_FLG	(varchar 1). Debit group pre-advise account limit exception. This column contains the following values: E (preadvise limit on debit) or O (override).
DBT_AMOUNT	(number 21.3). Debit dollar value of the transaction; mapped from the AMT field on the FTR screen. Cross-currency mode only.
DBT_CURRENCY	(varchar 3). Debit foreign currency code; mapped from the AMT field on the FTR screen. Cross-currency mode only.
DBT_RT_CODE	(varchar 6). Bank-assigned mnemonic that provides a shortcut for routing messages to one or more queues for printing; mapped from the RTC field on the FTR screen.
DBT_BNK_INF1	(varchar 35). First line of additional text to send with the transaction; mapped from the BANK TO BANK INFO field on the FTR screen.
DBT_BNK_INF2	(varchar 35). Second line of additional text to send with the transaction; mapped from the BANK TO BANK INFO field on the FTR screen.
DBT_BNK_INF3	(varchar 35). Third line of additional text to send with the transaction; mapped from the BANK TO BANK INFO field on the FTR screen.
DBT_BNK_INF4	(varchar 35). Fourth line of additional text to send with the transaction; mapped from the BANK TO BANK INFO field on the FTR screen.
DBT_BNK_INF5	(varchar 35). Fifth line of additional text to send with the transaction; also known as bank-to-bank information.

DBT_BNK_INF6	(varchar 35). Sixth line of additional text to send with the transaction; also known as bank-to-bank information.
DBT_SEC_SUB_ACCT	(varchar 61). Debit sub-account (in the case of a securities transaction).
DBT_ACC_PROD_CODE	(varchar 20). Debit account product codes.
DBT_TER_IDTYPE	(varchar 1). Debit tertiary account ID type; contains the following values: D (DDA) F (Foreign nostro account number) G (General ledger account number) P (CHIPS participant ID) V (Savings account number)
DBT_TER_IDKEY	(varchar 30). Debit tertiary account ID.
DBT_FEE_CODE	(varchar 8). Debit fee code.
DBT_RECON_REF	(varchar 12). Account ID of reconciliation account that will be debited for this transfer. This column, when sometimes used for affiliate processing, is the affiliate bank ID.
DBT_ACC_PARENT_CD	(varchar 10). Account code of parent account that will be debited for this transfer.
DBT_CUSTOMER_ID	(varchar 20). ID of customer address that is associated this transfer.
DBT_DRAFT_NUM	(varchar 16). Check or draft number.
SBK_IDTYPE	(varchar 1). ID type of the sending bank; mapped from the SEND fields on the FTR screen. This column contains the following values: A (Fedwire routing ABA number) B (Branch) C (CHIPS universal ID) E (Extended ID) P (CHIPS participant ID) S (S.W.I.F.T. address, 8 or 11 characters) T (Telex number) U (User-defined name) X (Telex answerback) Z (Cable address)
SBK_ID	(varchar 64). Sending bank's identifier; mapped from the SEND fields on the FTR screen.
SBK_NAME1	(varchar 35). First line of sending bank name and address; mapped from the SEND fields on the FTR screen.

Message Tables

SBK_NAME2	(varchar 35). Second line of sending bank name and address; mapped from the SEND fields on the FTR screen.
SBK_NAME3	(varchar 35). Third line of sending bank name and address; mapped from the SEND fields on the FTR screen.
SBK_NAME4	(varchar 35). Fourth line of sending bank name and address; mapped from the SEND fields on the FTR screen.
SBK_REF_NUM	(varchar 20). Sending bank's reference number for this message; mapped from the SEND fields on the FTR screen.
OBK_IDTYPE	(varchar 1). ID type of the ordering bank; mapped from the ORDER BNK field on the FTR screen. This column contains the following values: A (Fedwire routing ABA number) B (Branch) C (CHIPS universal ID) E (Extended ID) P (CHIPS participant ID) S (S.W.I.F.T. address, 8 or 11 characters) T (Telex number) U (User-defined name) X (Telex answerback) Z (Cable address)
OBK_ID	(varchar 64). Ordering bank's identifier; mapped from the ORDER BNK fields on the FTR screen.
OBK_NAME1	(varchar 35). First line of ordering bank name and address; mapped from the ORDER BNK fields on the FTR screen.
OBK_NAME2	(varchar 35). Second line of ordering bank name and address; mapped from the ORDER BNK fields on the FTR screen.
OBK_NAME3	(varchar 35). Third line of ordering bank name and address; mapped from the ORDER BNK fields on the FTR screen.
OBK_NAME4	(varchar 35). Fourth line of ordering bank name and address; mapped from the ORDER BNK fields on the FTR screen.
OBK_REF_NUM	(varchar 20). Ordering bank's reference number for this message; mapped from the ORDER BNK field on the FTR screen.
ORP_IDTYPE	(varchar 1). ID type of the originator account; mapped from the ORIG fields on the FTR screen. This column contains the following values: A (Fedwire routing ABA number) B (Branch) C (CHIPS universal ID)

	<p>E (Extended ID)</p> <p>P (CHIPS participant ID)</p> <p>S (S.W.I.F.T. address, 8 or 11 characters)</p> <p>T (Telex number)</p> <p>U (User-defined name)</p> <p>X (Telex answerback)</p> <p>Z (Cable address)</p> <p>Space (Foreign account)</p>
ORP_ID	(varchar 64). Originator identifier; mapped from the ORIG fields on the FTR screen.
ORP_NAME1	(varchar 35). First line of originator name and address; mapped from the ORIG fields on the FTR screen.
ORP_NAME2	(varchar 35). Second line of originator name and address; mapped from the ORIG fields on the FTR screen.
ORP_NAME3	(varchar 35). Third line of originator name and address; mapped from the ORIG fields on the FTR screen.
ORP_NAME4	(varchar 35). Fourth line of originator name and address; mapped from the ORIG fields on the FTR screen.
ORP_REF_NUM	(varchar 20). Originator's reference number for this transaction; mapped from the ORIG field on the FTR screen.
RECORD_EXPIRED	(number 23). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
INS_ADR_BNK_ID	(varchar 3). Instructing party's address bank ID.
INS_REL_ID	(number 10). Instructing party's address reference ID.
INS_IDTYPE	<p>(varchar 1). ID type of the instructing party; mapped from the SEND fields on the FTR screen. This column contains the following values:</p> <p>A (Fedwire routing ABA number)</p> <p>B (Branch)</p> <p>C (CHIPS universal ID)</p> <p>E (Extended ID)</p> <p>P (CHIPS participant ID)</p> <p>S (S.W.I.F.T. address, 8 or 11 characters)</p> <p>T (Telex number)</p> <p>U (User-defined name)</p> <p>X (Telex answerback)</p>

Message Tables

	Z (Cable address)
INS_ID	(varchar 64). Instructing party's ID.
INS_NAME1	(varchar 35). First line of the instructing party's name and address.
INS_NAME2	(varchar 35). Second line of the instructing party's name and address.
INS_NAME3	(varchar 35). Third line of the instructing party's name and address.
INS_NAME4	(varchar 35). Fourth line of the instructing party's name and address.
INS_REF_NUM	(varchar 16). Instructing party's reference number for this message.
DBT_SHORT_FLG	(varchar 1). Account short limit risk overdraft.
DBT_GROUP_SHORT_FLG	(varchar 1). Group short limit risk overdraft.
DBT_CROSS_SHORT_FLG	(varchar 1). Exceeded cross currency short limit.
DBT_EXCH_RATE	(number 29.11). Calculated exchange rate.
DBT_ADR_BNK_ID	(varchar 3). Debit party address bank ID.
DBT_REL_ID	(varchar 11). Debit party address reference ID.
DBT_ADR_CLASS	(varchar 6). Debit party address class.
DBT_ADR_CITY	(varchar 15). Debit party address city.
DBT_ADV_TYP	(varchar 3). Debit advice type.
DBT_RT_STATE	(varchar 1). Indicates whether the routing code instructions have been queued.
DBT_WIR_IDTYPE	(varchar 1). ID type for the debit wire key.
DBT_WIR_ID	(varchar 64). ID for the debit wire key.
DBT_DRAWDOWN_FLG	(varchar 1). Indicates whether the account drawdown exception is enabled.
DBT_PS_ELIG_FLG	(varchar 1). Pre/split eligibility.
DBT_ITEM_HOLD_FLG	(varchar 1). Indicates whether the risk item hold is overridden.
DBT_NOF_LOOKED_UP	(varchar 1). Indicates whether the NOF credit party has been looked up.
DBT_CURRENCY_SUSP_FLG	(varchar 1). Indicates whether the debit currency has been suspended.
DDA_BALANCE	(varchar 2). DDA balance.
BALANCE_RISK	(varchar 2). Balance risk.

SBK_ADR_BNK_ID	(varchar 3). Sending bank party address bank ID.
SBK_REL_ID	(varchar 11). Sending bank party address reference ID.
OBK_ADR_BNK_ID	(varchar 3). Originating bank party address bank ID.
OBK_REL_ID	(varchar 11). Originating bank party address reference ID.
ORP_ADR_BNK_ID	(varchar 3). Originating party address bank ID.
ORP_REL_ID	(varchar 11). Originating party address reference ID.
DBT_STATE	(varchar 3). State or province of the debit party.
DBT_POSTAL_CODE	(varchar 15). Postal code of the debit party.
DBT_RES_COUNTRY	(varchar 2). Resident country of the debit party.
DBT_REF_NUM	(varchar 16). Reference number of the debit party.
SBK_STATE	(varchar 3). State or province of the sending bank party.
SBK_POSTAL_CODE	(varchar 15). Postal code of the sending bank party.
SBK_RES_COUNTRY	(varchar 2). Resident country of the sending bank party.
OBK_STATE	(varchar 3). State or province of the ordering bank party.
OBK_POSTAL_CODE	(varchar 15). Postal code of the ordering bank party.
OBK_RES_COUNTRY	(varchar 2). Resident country of the ordering bank party.
ORP_STATE	(varchar 3). State or province of the originator party.
ORP_POSTAL_CODE	(varchar 15). Postal code of the originator party.
ORP_RES_COUNTRY	(varchar 2). Resident country of the originator party.
INS_STATE	(varchar 3). State or province of the instructing bank party.
INS_POSTAL_CODE	(varchar 15). Postal code of the instructing bank party.
INS_RES_COUNTRY	(varchar 2). Resident country of the instructing bank party.
RCA_STATE	(varchar 3). State or province of the receiving agent.
RCA_POSTAL_CODE	(varchar 15). Postal code of the receiving agent.
RCA_RES_COUNTRY	(varchar 2). Resident country of the receiving agent.

Message Tables

DBT_AUTH_REF	(varchar 34). Authorizing reference ID returned from the host during MPA processing.
DBT_VALUE_DATE	(date). Date on which funds are presumed to be removed, for interest purposes.
DBT_VALUE_DATE_CHG	(varchar 20). Degree of change enforcement for a non-business date.
DBT_VALUE_DATE_ADJ	(varchar 20). Forward or backward adjustment for the debit value date.
DBT_VALUE_DATE_SP	(varchar 20). Indicates special circumstances under which the debit value date must be replaced.
DBT_BOOK_DATE	(date). Book date, or when debit accounting is posted to the mainframe.
DBT_BOOK_DATE_CHG	(varchar 20). Degree of change enforcement for a non-business date.
DBT_BOOK_DATE_ADJ	(varchar 20). Forward or backward adjustment for the debit book date.
DBT_BOOK_DATE_SP	(varchar 20). Indicates special circumstances under which the debit book date must be replaced.
DBT_CAMEFROM	(varchar 20). Source where the debit party was found.
SBK_CAMEFROM	(varchar 20). Source where the sending bank party was found.
OBK_CAMEFROM	(varchar 20). Source where the ordering bank party was found.
ORP_CAMEFROM	(varchar 20). Source where the originator party was found.
INS_CAMEFROM	(varchar 20). Source where the instructing party was found.
RCA_CAMEFROM	(varchar 20). Source where the receiving agent was found.
DEBITSIDE_RESIDENCY	(varchar 2). Residence of the ultimate debit party, for example, the originating party, ordering bank, or debit party.
DBT_PROFILE_IDBANK	(varchar 3). Bank used for the debit party profile.
DBT_PROFILE_IDTYPE	(varchar 1). ID type for the debit party profile.
DBT_PROFILE_IDACC	(varchar 30). Account number for the debit party profile.
DBT_PROFILE_IDADR	(varchar 5). Address code for the debit party profile.
DBT_PROFILE_IDPAD	(varchar 1). Padding for the debit party profile.
SBK_PROFILE_IDBANK	(varchar 3). Bank used for the sending bank party profile.
SBK_PROFILE_IDTYPE	(varchar 1). ID type for the sending bank party profile.
SBK_PROFILE_IDACC	(varchar 30). Account number for the sending bank party profile.

SBK_PROFILE_IDADR	(varchar 5). Address code for the sending bank party profile.
SBK_PROFILE_IDPAD	(varchar 1). Padding for the sending bank profile.
DBT_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the debit party.
SBK_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the sending bank party.
OBK_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the ordering bank party.
ORP_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the originator party.
INS_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the instructing bank party.
RCA_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the receiving agent party.
DBT_BILAT_IDBANK	(varchar 3). Bank ID for the debit bi-lateral account.
DBT_BILAT_IDTYPE	(varchar 1). ID type for the debit bi-lateral account.
DBT_BILAT_IDKEY	(varchar 30). Account number for the debit bi-lateral account.
DBT_SEC_IDBANK	(varchar 3). Bank ID for credit secondary account.
DBT_SEC_IDTYPE	(varchar 1). ID type for credit secondary account.
DBT_SEC_IDKEY	(varchar 30). Account number for credit secondary account.
DBT_SEC_ACCTG_CUR	(varchar 3). Debit secondary accounting currency.
DBT_TER_IDBANK	(varchar 3). Debit tertiary accounting bank.
DBT_TER_ACCTG_CUR	(varchar 3). Debit tertiary accounting currency.
DBT_IBAN	(varchar 1). T = debit side has an IBAN
DBT_PRULE_COPY	(varchar 11). State of debit party processing rule copy into message.
SBK_PRULE_COPY	(varchar 11). State of sending bank party processing rule copy into message.
DBT_PROC_RULE_STATE	(varchar 9). Debit processing rule caching data.
SBK_PROC_RULE_STATE	(varchar 9). Sending bank processing rule caching data.
DBT_SPC_INST1	(varchar 35). Debit special instructions, line 1. Special instructions are used to recreate the existing full transaction report that lists all transactions for a specific business day in a standard MTS full message print format.

Message Tables

DBT_SPEC_INST2	(varchar 35). Debit special instructions, line 2. Special instructions are used to recreate the existing full transaction report that lists all transactions for a specific business day in a standard MTS full message print format.
DBT_SPEC_INST3	(varchar 35). Debit special instructions, line 3. Special instructions are used to recreate the existing full transaction report that lists all transactions for a specific business day in a standard MTS full message print format.
DBT_COMM_CBL_FLG	(varchar 1). C = Display REL account's charges.

Message Accounting Table (MESSAGE_ACCTG_T)

The Message Accounting table contains accounting instructions.

Column Descriptions

TRN_DATE (pk)	(date). Date from the TRN field; processing date on which the message was created.
TRN_NUMBER (pk)	(varchar 8). Unique serial number assigned to the message by REFserv (a background process in MTS that assigns the second half of the next consecutive transaction number) . The first transaction message of each processing day is 00000001.
ACCTG_IDTYPE (pk)	(varchar 1). Account ID type of underlying account to be debited or credited; correlates to the ACCTG_ACC_IDTYPE field in the MFE. This column contains the following values: <ul style="list-style-type: none">• D (DDA)• F (Foreign nostro account number)• G (General ledger account number)• P (CHIPS participant ID)• V (Savings account number)
ACCTG_ACCOUNT(pk)	(varchar 30). Account ID of underlying account to be debited or credited; correlates to the ACCTG_ACC_ID field of the MFE.
AMOUNT (pk)	(number 21.3). Transaction amount; mapped from the AMT field on the FTR screen.
STS_DBT_CDT (pk)	(varchar 1). Flag indicating whether the transaction is a debit or credit. This column contains the following values: C (credit). or D (debit). .
STS_CROSS_BANK (pk)	(varchar 1). Flag indicating whether any underlying cross-bank accounting has taken place to balance the transaction between the two banks involved in the transfer. This column contains the following values: Y (such accounting has taken place). or N (no such accounting has taken place). . Note: The underlying cross-bank accounts associated with each bank in the database are stored in the MTS Bank File (BNk) .

STS_PRIMARY (pk)	<p>(varchar 1). Flag indicating whether accounting is primary, secondary, or tertiary. This column contains the following values:</p> <ul style="list-style-type: none"> • P (Primary; dollar amount of actual transaction and related cross-bank accounting). • S (Secondary accounting offset for commission charges, for example; or, in the case of Fed Funds purchases and sales, when interest is split from principal). • T (Tertiary accounting offset, usually cable charges). <p>For example, if a commission charge of \$5.00 and a cable charge of \$3.00 were assessed on a \$100.00 wire, each component of the transaction would be assigned an STS_PRIMARY flag as follows: \$100.00 wire P, \$5.00 commission S, \$3.00 cable T.</p>
TRN_TIMESTAMP (pk)	<p>(number 23). Date and time when this Message File Extract record was written; correlates to the NOW_TIMESTAMP field of the Message Extract File (MFE). .</p>
PERIOD	<p>(date). Date on which the data export occurred.</p>
PROC_DATE	<p>(date). Text version of the PERIOD field in yymmdd format; current processing date. The Proc Date is the same in each record of a Message File Extract file.</p>
BANK (pk)	<p>(varchar 3). Bank identifier of the bank where this message was created; mapped from the BANK_ID field in the FTR screen.</p>
LOC	<p>(varchar 6). Site-defined location (within bank) where this message was created. MTRANS is usually the standard value. No location is associated with incoming network transactions.</p>
VALUE_DATE	<p>(date). Current, future, or back-dated transaction date; mapped from the VAL field on the FTR screen.</p>
INST_DATE	<p>(date). Instruction received date. Date of receipt; set by input mapper of the communication link.</p>
TRAN_TYPE	<p>(varchar 3). Transaction type; mapped from the TYP field on the FTR screen. This column contains the following values:</p> <ul style="list-style-type: none"> • DEP (Deposit) • DFA (Draft advice) • DFT (Draft) • DRR (Drawdown request) • DRW (Drawdown response) • FFB (Fed Funds bought) • FFR (Fed Funds returned) • FFS (Fed Funds sold) • FTR (Funds transfer) • NON (Nonaccounting)

Message Tables

REPETITIVE_ID	(varchar 48). Unique identifier assigned to the repetitive template MTS uses to generate a recurring outgoing transaction. The Repetitive ID is a combination of multiple fields. The number is mapped from the RPT# field on the FTR screen. The debit party is mapped from the BNK_ID, DBT_IDTYPE, and DBT_ID fields on the FTR screen.
SOURCE_CD	<p>(varchar 3). Source code; mapped from the SRC field on the FTR screen. This column contains the following values:</p> <ul style="list-style-type: none">• ADJ (Adjustment)• ADM (Administrative message entry function)• CHP (CHIPS)• CMS (Cash management system)• DFM (Due from monitor)• ENT (Payment entry function)• FED (Fedwire)• INT (Internal transfer)• LTC (Letter requiring callback)• LTR (Letter)• MEM (Memo past function)• MIS (Miscellaneous, for example, cash management system or remote batch entry)• MTW (Money Transfer for Windows)• OFL (Offline)• PHN (Phone transfer initiated outside bank)• SOD (Start-of-day balance load)• STO (Standing order function)• SWF (S.W.I.F.T.)• WIR (Telex)• Any user-defined INTRTL source.
INSTR_ADV_TYPE	<p>(varchar 3). Advice method indicated by sender for the credit party; mapped from the ADV field on the FTR screen. This column contains the following values:</p> <ul style="list-style-type: none">• CHK (Check)• CHP (CHIPS)• FED (Fedwire)• LTR (No primary advice; secondary confirmation only)• PHH (Phone hold)• PHN (Phone)• SWF (S.W.I.F.T.)• TKT (Online ticket)• WIR (Telex)• Any user-defined INTRTL advice

TYPE_CD	(varchar 2). Type code mapped from the MTP field on the Funds Transfer (FTR) screen. Outgoing Federal Reserve message type; correlates to the MSG_TYPE_TYP field in the Message Extract File (MFE).
SUBTYPE	(varchar 2). Subtype code mapped from the MTP field on the Funds Transfer (FTR) screen. Outgoing Federal Reserve message subtype code; correlates to the MSG_TYPE_SUBTYP field in the Message Extract File (MFE).
IN_TYPE_CD	(varchar 2). Incoming Federal Reserve, SWIFT, or CHIPS message type code; correlates to the INMSG_TYPE_TYP field in the MFE.
IN_SUBTYPE	(varchar 2). Incoming Federal Reserve, SWIFT, or CHIPS message subtype code; correlates to the INMSG_TYPE_SUBTYP field in the MFE.
DBT_CHRG	(varchar 1). Flag indicating whether charges apply for debit; mapped from the CHG:DB field on the FTR screen. Values are bank-defined, determined from user-defined configuration routines for fee charges.
CDT_CHRG	(varchar 1). Flag indicating whether charges apply for credit; mapped from the CHG:CD field on the FTR screen. Values are bank-defined, determined from user-defined configuration routines for fee charges.
COMMISSION	(varchar 1). Flag indicating whether charges apply for commission; mapped from the CHG:COM field on the FTR screen. Values are bank-defined, determined from user-defined configuration routines for commission charges.
CBL_CHARGE	(varchar 1). Flag indicating whether charges are to be applied for cable; mapped from the CHG:CAB field on the FTR screen. Values are bank-defined, determined from user-defined configuration routines for cable charges.
WIRE_TYPE	<p>(varchar 3). Wire type; used in end-of-day reporting. Site-specific combination of SRC and ADV_TYPE; determined by configuration file settings. This column contains the following values:</p> <ul style="list-style-type: none"> • FWI (Fed wire incoming) • FWO (Fed wire outgoing) • SWI (S.W.I.F.T. wire incoming) • SWO (S.W.I.F.T. wire outgoing) • TWI (Telex wire incoming) • TWO (Telex wire outgoing) • IBT (Interbank transfer) • Blank (Default)
IDTYPE	<p>(varchar 1). Debit or credit party's ID type; mapped from the first part of the DBT or CDT field (left of the slash). on the FTR screen. This column contains the following values:</p> <ul style="list-style-type: none"> • A (Fedwire routing ABA number) • B (Branch code) • C (CHIPS user ID)

Message Tables

	<ul style="list-style-type: none"> • D (DDA) • E (Extended ID) • F (Foreign nostro account number) • G (General ledger account number) • I (Bank lookup key) • K (Customer ID number) • M (Private leased line network MAC code) • N (MTS mnemonic name) • P (CHIPS participant ID) • R (ACI Worldwide system address reference number) • S (S.W.I.F.T. address, 8 or 11 characters) • T (Telex number) • U (User-defined name) • V (Savings account number) • X (Telex answerback) • Z (Cable address)
ID	<p>(varchar 64). Debit or credit party's ID; mapped from the second part of the DBT/CDT field (right of the slash) on the FTR screen.</p> <p>Note:</p> <p>In a cross-bank database, the field begins with the string BBB followed by a colon (BBB:). . . , where BBB represents the Bank ID.</p>
NAME1	(varchar 35). Debit or credit party's name, line 1; associated with the IDTYPE and ID columns.
NOF_FLAG	(varchar 1). If the party is not on file, the value in this column is an asterisk (*). . . , otherwise, it is blank.
TRAN_NAME_C1	(varchar 8). Debit or credit transaction code name C1; site-defined code resides in the ACI Worldwide configuration files.
TRAN_NAME_C2	(varchar 8). Debit or credit transaction code name C2; site-defined code resides in configuration files.
TRAN_NAME_C3	(varchar 8). Debit or credit transaction code name C3; site-defined code resides in configuration files.
TRAN_NAME_C4	(varchar 8). Debit or credit transaction code name C4; site-defined code resides in configuration files.
TRAN_NAME_C5	(varchar 8). Debit or credit transaction code name C5; site-defined code resides in configuration files.
TRAN_NAME_C6	(varchar 8). Debit or credit transaction code name C6; site-defined code resides in configuration files.

TRAN_NAME_C7	(varchar 8). Debit or credit transaction code name C7; site-defined code resides in configuration files.
TRAN_NAME_C8	(varchar 8). Debit or credit transaction code name C8; site-defined code resides in configuration files.
TRAN_NAME_C9	(varchar 8). Debit or credit transaction code name C9; site-defined code resides in configuration files.
TRAN_NAME_C10	(varchar 8). Debit or credit transaction code name C10; site-defined code resides in configuration files.
DEPT	(varchar 20). Bank-defined department where the debit or credit account is assigned; mapped from the DEPT field on the FTR screen.
SERIAL	(varchar 10). Serial portion of the test algorithm for telex messages.
ACC_CLASS	(varchar 6). Site-defined account class column; mapped from the CLASS field on the Relationship File (REL). Account File Screen 1.
ADR_COUNTRY	(varchar 2). Address' country code; mapped from REL. Blank defaults to US. Correlates to the ADR_TYPE_COUNTRY field in the MFE.
ADR_TYPE	<p>(varchar 1). Address' relation type from the Address File Screen 1 in REL. Correlates to the ADR_TYPE_TYP field in the MFE. This column contains the following values:</p> <ul style="list-style-type: none"> • B (Bank) • C (Corporate) • I (Internal)
ADR_SUBTYP	(varchar 3). Address' relation subtype from the Address File Screen 1 in REL. Correlates to the ADR_TYPE_SUBTYP field in the MFE. Refer to the following table for possible values:

SEC_ACCTG_FLG	(varchar 16). Secondary accounting flag.
SEC_ACCTG_AMT	(number 21.3). Secondary accounting amount.
OFFSET_IDTYPE	<p>(varchar 1). Offset ID type. This column contains the following values:</p> <ul style="list-style-type: none"> • D (DDA) • F (Foreign nostro account number) • G (General ledger account number) • P (CHIPS participant ID) • V (Savings account number)
OFFSET_ID	(varchar 64). Debit or credit party's account ID. This column must be left-justified and filled with trailing blanks. In a cross-bank database, column begins with the string BBB:, where BBB represents the Bank ID.

Message Tables

OFFSET_NAME1	(varchar 35). Offset debit or credit party's name line 1.
FUNDS_TYPE	(varchar 1). Type of funds; mapped from the FNDS field on the FTR screen. This column contains the following values: N (next day funds) or S (same day funds).
SBK_REF_NUM	(varchar 20). Sending bank's reference number for this transaction; mapped from the SNDR REF NUM field on the FTR screen.
OBK_REF_NUM	(varchar 20). Originating bank's reference number for this transaction; mapped from the ORDER BNK field on the FTR screen.
ORP_REF_NUM	(varchar 20). Originating party's reference number for this transaction; mapped from the REF NUM field on the FTR screen.
ASC_AMT	(varchar 18). ASCII representation of the debit or credit accounting amount.
ASC_SEC_AMT	(varchar 18). ASCII representation of the debit or credit secondary accounting amount.
REC_TYPE	(varchar 1). Record type. A record can have multiple sets of entries in the Accounting File. The first set is considered the normal set and contains a P in this column. This set is written to the MFE_OUTQ by PAYADV processing. The second set is written to the MFE_OUTQ by CANCELSUBS when a message is cancelled. This column contains the following values: P (payment posting) or C (payment cancellation). .
LAST_ADDR	(varchar 35). Last Address.
DBT_AUTH_FLAG	(varchar 1). Debit authorization flag set by MPA or DDA Post (offline MTS functions that communicate with the mainframe to determine authorization before a transaction posts).
CDT_RECON_REF	(varchar 12). Credit reconciliation reference number.
DBT_RECON_REF	(varchar 12). Debit reconciliation reference number.
CUR_CODE	(varchar 3). Currency code involved in the transaction.
RECORD_EXPIRED	(number 23). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Message Destination Table (MESSAGE_DEST_T)

The Message Destination table describes the routing of a message and provides a list of the parties notified of a transfer.

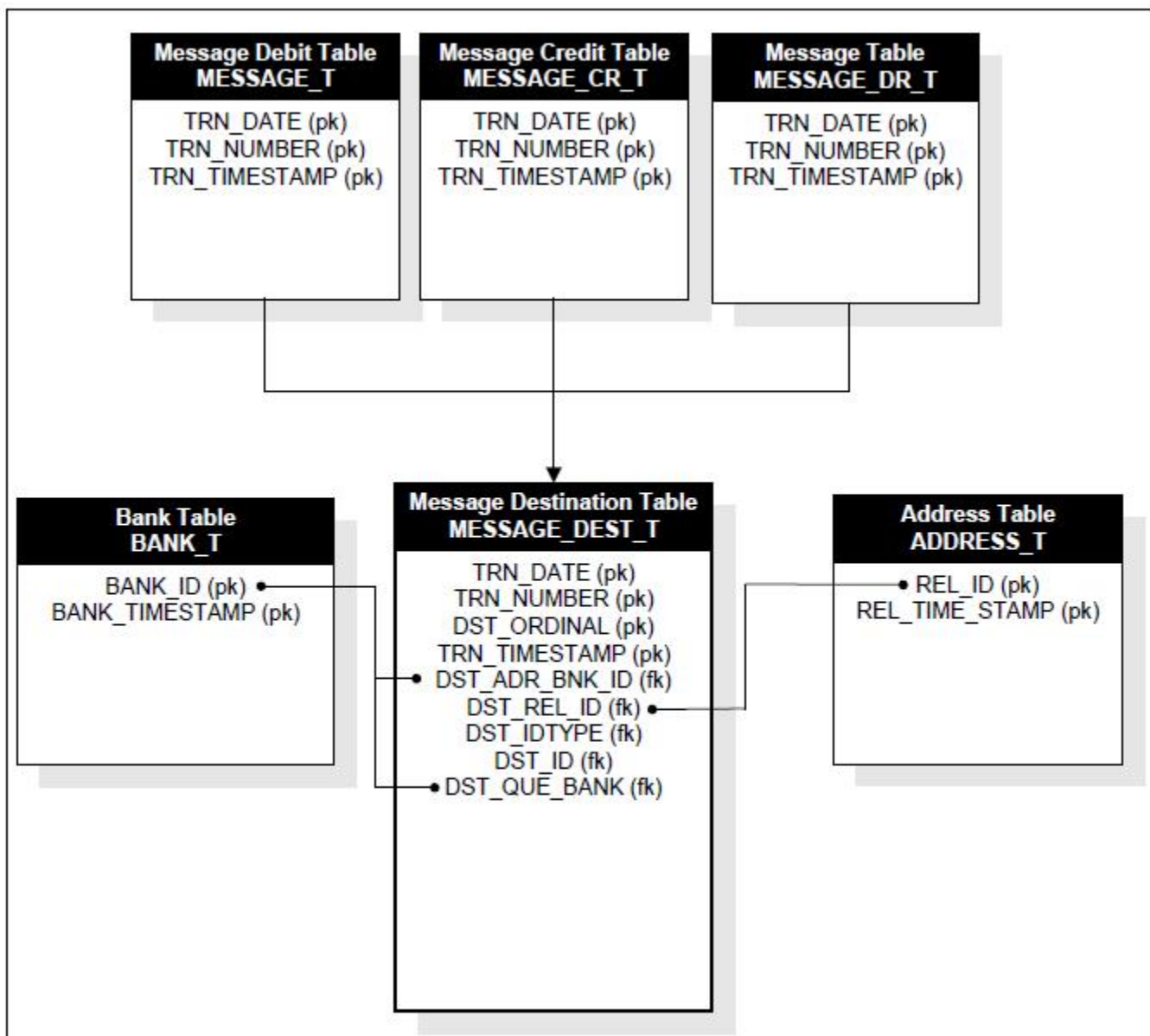
Table Relationships

The DST_ADR_BANK_ID and DST_QUE_BANK columns in the Message Destination table relate to BANK_ID in the Bank table.

The DST_REL_ID column in the Message Destination table relates to REL_ID in the Address table.

Relationship diagram

This relationship diagram shows the relationship between the Message Destination table, its parent tables, and the tables related to it through foreign keys.



Column Descriptions

TRN_DATE (pk)	(date). Date from the TRN field; processing date on which the message was created.
TRN_NUMBER (pk)	(varchar 8). Unique serial number assigned to the message by REFSERV (a background process in MTS that assigns the second half of the next consecutive transaction number). The first transaction message of each processing day is 00000001.
DST_ORDINAL (pk)	(number 5). Ordinal of this destination within the sequence of destinations for this message.
TRN_TIMESTAMP (pk)	(number 23). Date and time when this Message File Extract record was written; correlates to the NOW_TIMESTAMP field of the MFE.
ALTERNATE_ACCT	(varchar 12). Alternate account number.
COST_CENTER	(varchar 10). Cost center charged for the transaction.
PARAM_1	(varchar 8). Identifies a valid charging modifier.
PARAM_2	(varchar 8). Identifies a valid charging modifier.
IDENTIFIER	(varchar 8). Identifies the bank to which the account is related.
DST_ADR_BNK_ID (fk)	(varchar 3). Bank ID under which the address text was found in the REL file.
DST_REL_ID (fk)	(varchar 11). Rel ID, if the destination address came from the REL file.
DST_OVR	(varchar 1). Override flag, an asterisk(*) indicates that this address is not on file. If DST_OVR is * (not on file) , an MTS auxiliary information file may have been used to determine the address.
DST_IDTYPE (fk)	(varchar 1). ID type of the destination account, can include these values: A (Fedwire routing ABA number) B (Branch code) C (CHIPS user ID) E (Extended ID) K (Customer ID number) M (Private leased line network MAC code) P (CHIPS participant) S (S.W.I.F.T. address, 8 or 11 characters) T (Telex number) U (User-defined name) V (Savings account number)

	X (Telex answerback) Z (Cable address)
DST_ID (fk)	(varchar 64). Destination ID; standard address (party) key, used to find this address in the REL file.
DST_TYPE	(varchar 3). Primary wire service/delivery type. This column contains the following values: CHP (CHIPS) ENQ (Other processing queue) FAX FED (Fedwire) ITX (Incoming telex) MAC (Machine authenticated code) PRT (Printer) PVT (Private carrier) SAF (Telex store and forward) SWF (S.W.I.F.T.) TLX (Telex) WTX (Outgoing telex)
DST_CARRIER	(varchar 3). Destination carrier. Possible values include RCA, WUI, TRT, ATT, MCI, and others.
DST_AREA_CODE	(varchar 4). Destination area code.
DST_DIAL	(varchar 16). Destination telephone or telex number.
DST_SIGN	(varchar 1). This column is used for MAC (machine code) or PVT (private) carriers; it will contain a plus sign (+) or appear blank.
DST_MAC	(varchar 10). Code used for MAC (machine code) deliveries.
DST_ANS	(varchar 20). Answerback character string used for telex deliveries.
DST_NAME1	(varchar 35). First line of correspondent or cable name and address.
DST_CAB2	(varchar 35). Second line of correspondent or cable name and address.
DST_CAB3	(varchar 35). Third line of correspondent or cable name and address.
DST_CAB4	(varchar 35). Fourth line of correspondent or cable name and address.
DST_CITY	(varchar 15). City portion of correspondent or cable name and address.
DST_COUNTRY	(varchar 2). Country portion of correspondent or cable name and address.

Message Tables

DST_QUE_BANK (fk)	(varchar 3). Bank identifier portion of fully-specified MTS queue name when message is to be delivered to an MTS printer queue (PRT) or other processing queue (ENQ). (This is the bank part of a three-part queue name.).
DST_QUE_LOC	(varchar 6). Location portion of fully specified MTS queue name when message is to be delivered to an MTS printer queue (PRT) or other processing queue (ENQ). (This is the location part of a three-part queue name.). .
DST_QUE_NAME	(varchar 12). Fully specified MTS queue name when message is to be delivered to an MTS printer queue (PRT) or other processing queue (ENQ). (This is the queue part of a three-part queue name.).
DST_ROUTE_ID	(varchar 64). Appropriate routing ID when delivery is via FED, S.W.I.F.T, or PVT (private). This column contains the following values: ABA routing number, or a S.W.I.F.T. ID.
DST_DEPARTMENT	(varchar 20). Department portion of correspondent or cable name and address.
DST_RT_CODE	(varchar 6). Routing code that specifies one of a set of pre-defined printer groups when delivery is via printer.
DST_CHAR_COUNT	(varchar 11). Number of characters in the outgoing message text stored in the destination set. This text varies according to the delivery mechanism. Characters are counted from the start of the header to the end of the text.
DST_MULTI_SEQ	(varchar 6). Sequence number within group.
DST_MULTI_START	(varchar 6). Line number of text within group.
DST_MULTI_COUNT	(varchar 6). Number of lines within group.
DST_OUT_TEST	(varchar 1). Flag indicating whether an outgoing testword is required (Y) or not (N).
DST_PDM_FLAG	(varchar 1). Flag indicating whether this message may have already been delivered to this recipient without a proper acknowledgment. This column contains the following values: Y (possibly duplicated message) or N or blank (not duplicated).
DST_STATE	(varchar 1). Indicates whether message was delivered. This column contains the following values: D (Delivered) Blank (Not processed) Q (Queued for delivery) S (Selected for delivery) F (Selected and formatted for delivery) N (Instruction not acknowledged (NAK'd).)
DST_ATTEN	(varchar 60). Attention line; mapped from ATTN field on WRP (wire repair), FAX, and other screens.

DST_SEC_TYPE	(varchar 3). Secondary or alternate wire service/delivery type; cannot be the same carrier as the primary delivery.
DST_SEC_AREA_CODE	(varchar 4). Secondary area code.
DST_SEC_DIAL	(varchar 16). Secondary telephone or telex number.
DST_SEC_SIGN	(char 1). Secondary column used for MAC (machine code) or PVT (private) carriers, it will contain a plus sign (+) or will be blank.
DST_SEC_MAC	(varchar 10). Secondary code used for MAC (machine code). deliveries.
DST_SEC_ANS	(varchar 20). Secondary answerback character string used for telex deliveries.
DST_SEC_NAME1	(varchar 35). Secondary correspondent or cable name and address.
DST_SEC_CAB2	(varchar 35). Secondary correspondent or cable name and address.
DST_SEC_CAB3	(varchar 35). Secondary correspondent or cable name and address.
DST_SEC_CAB4	(varchar 35). Secondary correspondent or cable name and address.
DST_ACK_STATE	(varchar 1). Delivery acknowledged status. Reserved for future use.
DST_NAK_STATE	(varchar 1). Delivery unacknowledged status. Reserved for future use.
DST_DELIVERY_FLG	(varchar 1). Delivery notification flag. Reserved for future use.
DST_ADR_LOCATION	(varchar 6). Site-defined location (within a bank) under which the address text was found in the REL file.
DST_3RD_PARTY_TST	(varchar 1). Destination column that contains 3rd-party test information. Reserved for future use.
DST_FORMAT	(varchar 4). Indicates the form name or letterhead form name used by a carrier when creating a fax. Reserved for custom use.
DST_MSGTYPE	(varchar 4). Identifies the message type; mapped from the REL file.
DST_CONN_TIME	(varchar 5). Telex charges for the amount of time you connect to the network.
DST_CHARGES	(varchar 7). Actual telex charges for the transaction.
DST_CARRIER_REF	(varchar 4). Actual charges from the telex carrier.
DST_TER_IDTYPE	(char 1). Third-party account ID type. This column contains the following values: D (DDA) F (Foreign nostro account number) G (General ledger account number) P (CHIPS participant ID)

Message Tables

	V (Savings account number)
DST_TER_ID	(varchar 14). Third-party account ID.
DST_TERTIARY_DEPT	(varchar 14). Tertiary department.
DST_MSG_COND	(number 10). Specifies that a message use a specific destination according to the message type.
DST_MSG_TYPE	(varchar 4). Destination message type.
DST_SRC_COND	(number 10). Specifies that a message came from a specific source according to the message type.
DST_SRC_ID	(varchar 3). Destination source ID for a message.
DST_TIME	(date). Time at which message was transmitted to the destination.
DST_OUTGOING_REF	(varchar 80). FED, S.W.I.F.T., or CHIPS external reference number for the message.
DST_TEST_DATE	(varchar 8). Date on which testing occurs for messages sent to a specific destination.
RECORD_EXPIRED	(number 23). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
DST_FIN_COPY_ID	(varchar 3). FIN copy service ID.
DST_119_HDR_DATA	(varchar 80). Text that appears in the SWF header on line 3, field 119.
DST_CASC_NAME	(varchar 33). Cascade routing.
DST_SRV_MSG_FLAG	(varchar 1). Indicates whether to send a service message (notification) to the third party in this data set.
DST_INC_HDR_FLAG	(varchar 1). Indicates whether to include a line of descriptive text in the message's header text.
DST_COPIES	(varchar 1). Indicates the number of copies to deliver. This applies to LTR (letters) only.
DST_NO_ACCT_STMT	(varchar 1). Indicates whether to send a letter confirmation.
DST_PRECEDENCE	(varchar 1). Indicates whether the destination has precedence over the unit confirmation. Y = Yes, the destination takes precedence.
DST_ORIGIN	(varchar 1). Determines how the destination was determined.
DST_113_HDR_DATA	(varchar 4). SWIFT header 3 field 113.

DST_EMAIL_ADDRESS	(varchar 80). E-mail address of the destination.
DST_TYPE_INDICATOR	(varchar 20). Type of destination set, indicating the method of advising.

Message History Table (MESSAGE_HIST_T)

The Message History table provides history information for each message received.

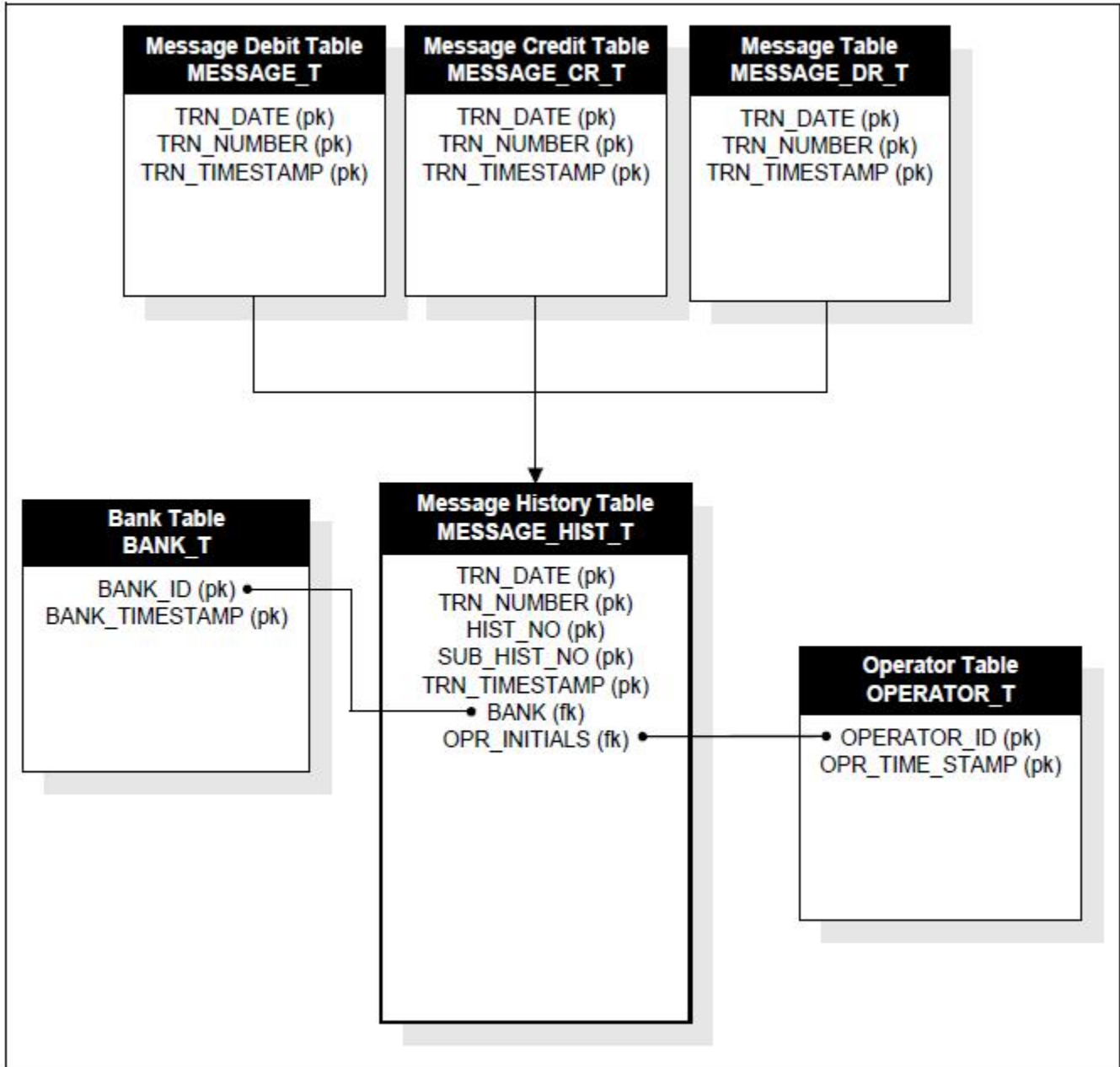
Table Relationships

The BANK column in the Message History table relates to BANK_ID in the Bank table.

The OPR_INITIALS column in the Message History table relates to OPERATOR_ID in the Operator table.

Relationship diagram

This relationship diagram shows the relationship between the Message History table, its parent tables and the tables related to the Message History table by foreign keys.



Column Descriptions

TRN_DATE (pk)	(date). Date from the TRN field; processing date when the message was created.
---------------	--

TRN_NUMBER (pk)	(varchar 8). Unique serial number assigned to the message by REFSEV (a background process in MTS that assigns the second half of the next consecutive transaction number). The first transaction message of each processing day is 00000001.
HIST_NO (pk)	(varchar 4). Ordinal to count the sequence number of an entry in MSG_HIST_SEQ.
SUB_HIST_NO (pk)	(varchar 4). Ordinal to count the sequence number of an entry in SUB_HIST_SEQ.
TRN_TIMESTAMP (pk)	(number 23). Date and time when this record was written.
ENTRY_TYPE	(varchar 3). 3-letter type code for this history entry.
BANK (fk)	(varchar 3). Bank ID associated with the message sub-history.
LOC	(varchar 6). Site-defined location (within bank) where this message was created. MTRANS is usually the standard value. No location is associated with incoming network transactions.
QUE_LINE_ID	(varchar 12). Delivery line, queue name, or log name.
DATE_TIME	(varchar 16). Date and time of the queue this record came from. This column may be blank.
SEQUENCE_NO	(varchar 7). Sequence number on communications line or printer.
DETAILS	(varchar 80). Free text additional memo, if any.
OPR_INITIALS (fk)	(varchar 10). Operator ID associated with the history event.
AMOUNT	(number 21.3). Value of the transaction; mapped from the AMT field on the FTR screen, if present.
MSG_INFO	(varchar 80). MIR (message input reference). record, GEN ID, IMAD, SSN, etc.
RECORD_EXPIRED	(number 23). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Message Queue Table (MESSAGE_QUEUE_T)

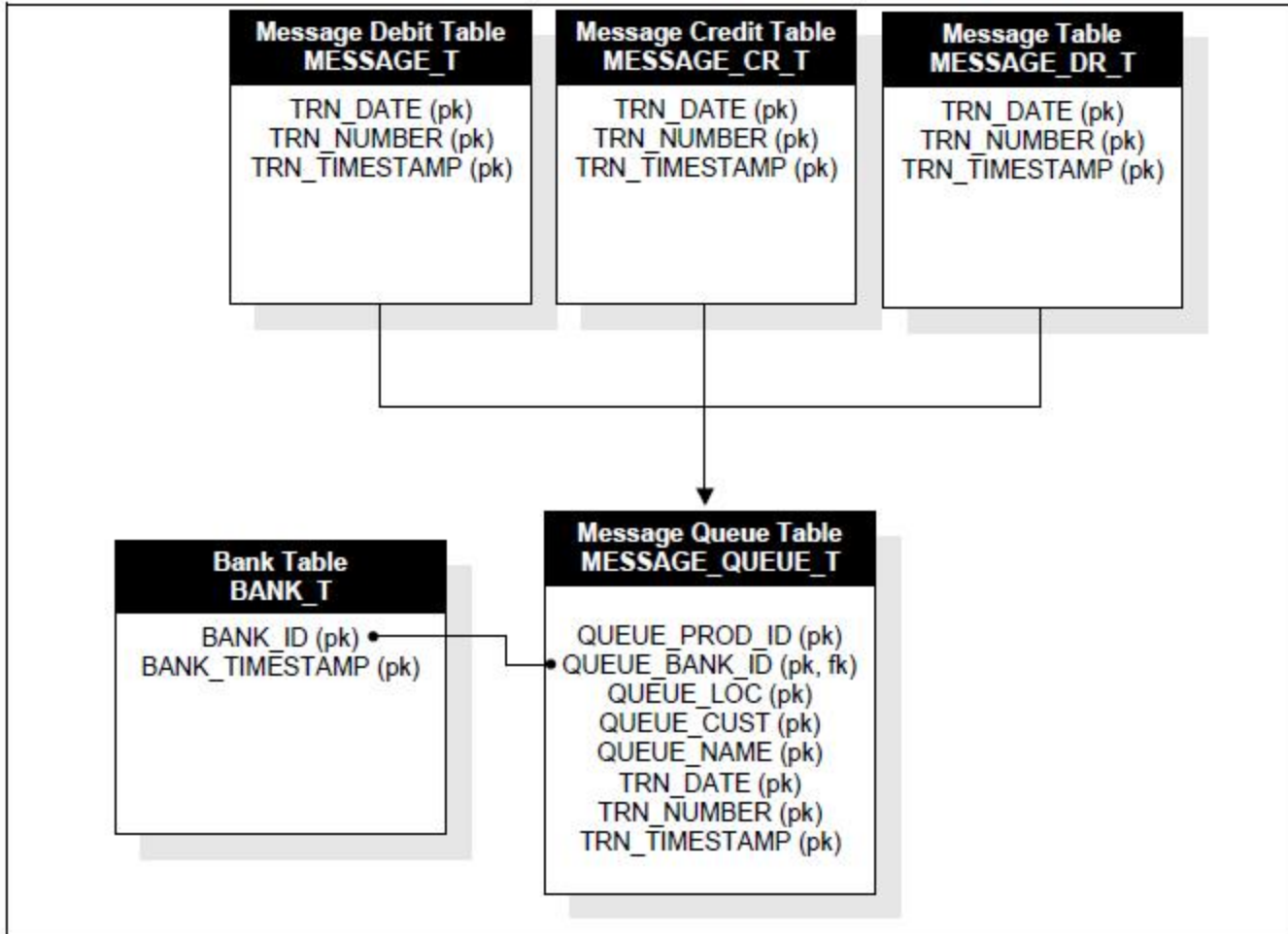
The Message Queue table shows messages that end the day on the future queue.

Table Relationships

The BANK column in the Message Queue table relates to BANK_ID in the Bank table.

Relationship Diagram

The relationship diagram on the next page shows the relationship between the Message Queue table, its parent tables and the table related to it by foreign keys.



Column Descriptions

QUEUE_PROD_ID (pk)	(varchar 3). Product ID.
QUEUE_BANK_ID (pk, fk)	(varchar 3). Bank ID.
QUEUE_LOC (pk)	(varchar 6). Queue location ID.
QUEUE_CUST (pk)	(varchar 20). Customer ID.
QUEUE_NAME (pk)	(varchar 33). Queue name.

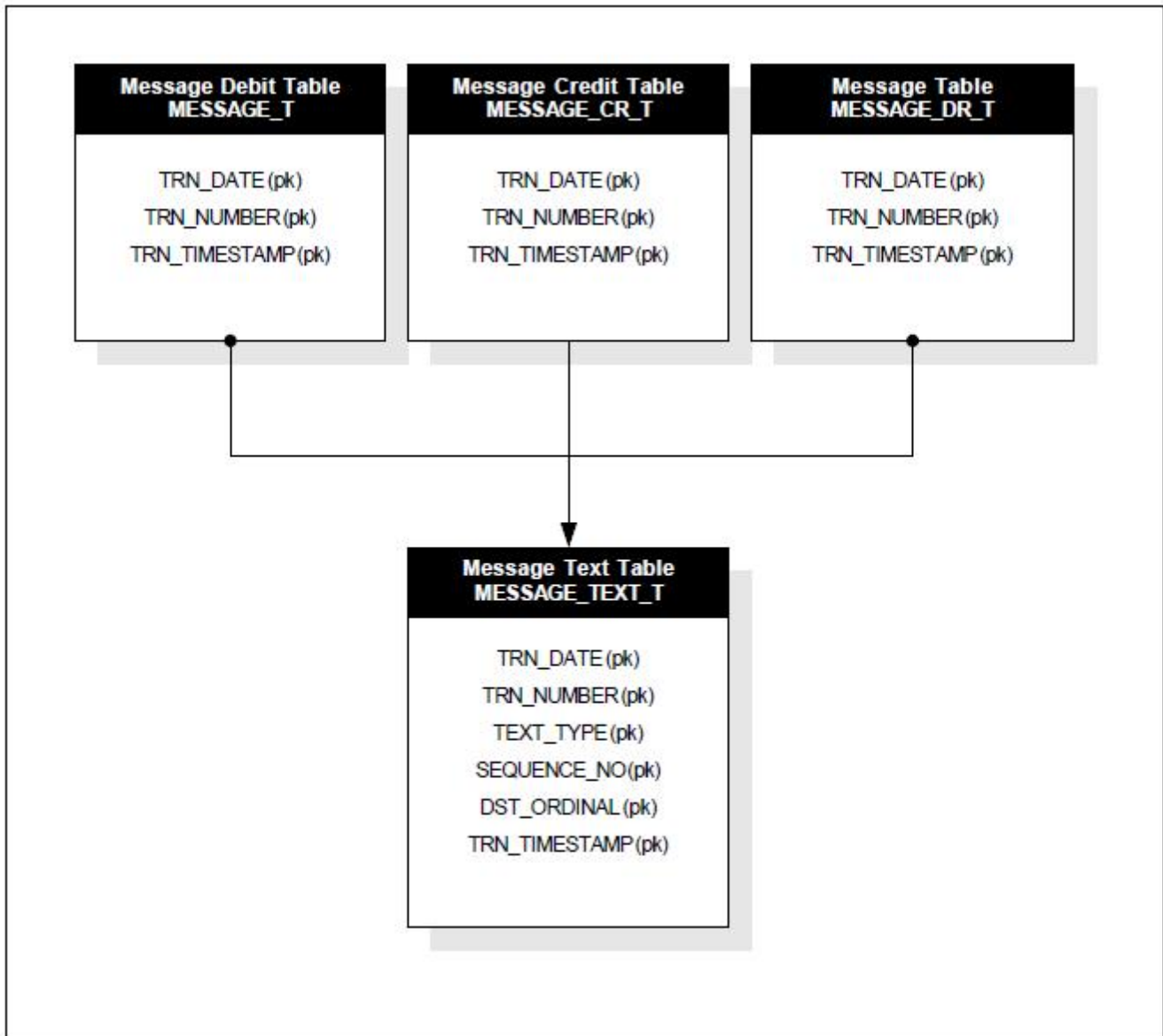
TRN_DATE (pk)	(date). Date from the TRN field; processing date on which the message was created.
TRN_NUMBER (pk)	(varchar 8). Unique serial number assigned to the message by REFSERV (a background process in MTS that assigns the second half of the next consecutive transaction number). The first transaction message of each processing day is 00000001.
TRN_TIMESTAMP (pk)	(date). Date and time when this record was written.
RECORD_EXPIRED	(varchar 16). Date and time at which this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
FUTURE_DATE	(date). Future date.
RLS_ACTION	(varchar 25). The reason for putting the message on the future queue.

Message Text Table (MESSAGE_TEXT_T)

The Message Text table provides the raw text received when a message arrives in MTS. This table does not have field code expansion and applies to incoming messages only.

Relationship Diagram

This relationship diagram shows the relationship between the Message Text table and its parent tables.



Column Descriptions

TRN_DATE (pk).	(date). Date from the TRN field; processing date on which the message was created.
----------------	--

TRN_NUMBER (pk)	(varchar 8). Unique serial number assigned to the message by REFSERV (a background process in MTS that assigns the second half of the next consecutive transaction number). The first transaction message of each processing day is 00000001.
TEXT_TYPE (pk)	(varchar 1). Identifies the type of text: I (Incoming) O (Primary Outgoing) S (Secondary Outgoing)
SEQUENCE_NO (pk)	(varchar 10). Message text sequence number.
DST_ORDINAL	(number 5). For formatted_text, the corresponding dst_seq ordinal.
TRN_TIMESTAMP (pk)	(number 23). Date and time when this record was written.
MESSAGE_TEXT	(varchar 80). Message text as received.
RECORD_EXPIRED	(number 23). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Message Related Amount Table (MESSAGE_REL_AMT_T)

The Message Related Amount table contains amount type information related to a message.

It is populated via a dump of the RELATED_AMOUNT_SEQ from the MSG_UNION. MESSAGE_REL_AMT_T can hold information on various amount types, depending on the value of the AMT_CODEWORD field in the sequence.

Primary Keys

The MESSAGE_REL_AMT_T table contains the following primary keys:

- RN_DATE
- TRN_NUMBER
- SEQUENCE_NO
- TRN_TIMESTAMP

Column Descriptions

TRN_DATE (pk)	(date). Date from the TRN field; processing date on which the message was created.
TRN_NUMBER (pk)	(varchar 8). Unique serial number assigned to the message by REFSESV (a background process in MTS that assigns the second half of the next consecutive transaction number). The first transaction message of each processing day is 00000001.
SEQUENCE_NO (pk)	(varchar 10). Message text sequence number.
TRN_TIMESTAMP (pk)	(number 23). Date and time when this record was written.
AMT_CODEWORD	(varchar 80). Codeword that specifies the type of amount. Values are: Fee Instructed Amount Multiple Amounts Receivers Charge Reg Senders Charge
AMOUNT	(number 21.3). Amount, of the particular type, in the party currency.
CURR	(varchar 3). Party currency.
MEMO	(varchar 80). Memo area for the related amount.
RECORD_EXPIRED	(number 23). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Message Returned Table (MESSAGE_RETURN_T)

The Message End of Day table contains a list of messages that were returned.

Column Description

TRN_DATE (pk)	(date). The period date when the message came into being.
TRN_NUMBER (pk)	(number 8). Transaction reference sequence number. The first transaction message of each processing day is 00000001.
MEMO	(varchar 80). Descriptive text displayed in address account sequence.
RMSG_DATE	(date). Date the message was returned.

RMSG_TRN_DATE	DATE
RMSG_TRN_NUMBER	NUMBER(8)
RMSG_REF	VARCHAR2(35)
RMSG_CSTMREF	VARCHAR2(35)

Message Processing Rules Tables

MTS uses the Relationship File (REL) to access instructions for processing a customer's messages.

When an incoming message contains an address on file in REL, MTS uses the information in this file to determine how to process the message. The Message Processing Rules Tables contain the instructions for processing messages.

Primary Keys

All of the Message Processing Rules tables have the following primary keys in common:

- TRN_DATE
- TRN_NUMBER
- PR_SEQ_NUM
- TRN_TIMESTAMP

Parent/child relationships

The Message Processing Rule table (MESSAGE_PR_T) is the parent to the following tables:

- Message Processing Rules Match Criteria table (MESSAGE_PR_MCH_T)
- Message Processing Rules Parameters table (MESSAGE_PR_PRM_T)
- Message Processing Rules Parameter Values table (MESSAGE_PR_PVL_T)

Message Processing Rules Table (MESSAGE_PR_T)

The Message Processing Rules table stores a list of instructions used to process customer's messages. The instructions, called processing rules, are associated with address, channel, or profile relationships.

Primary Keys

The MESSAGE_PR_T table contains the following primary keys:

- TRN_DATE
- TRN_NUMBER
- PR_SEQ_NUM
- TRN_TIMESTAMP

Column Descriptions

TRN_DATE (pk)	(date). Period date when the message came into being.
TRN_NUMBER (pk)	(number 8). Transaction reference sequence number. The first transaction message of each processing day is 00000001.
PR_SEQ_NUM (pk)	(number 10). Processing rule sequence number.
TRN_TIMESTAMP (pk)	(number 23). Date and time when the record was written.
PR_ORDINAL	(number 10). Processing rule ordinal.
PR_LEVEL	(varchar 20). Processing rule level.
PR_SOURCE	(varchar 25). Processing rule source.
PR_SOURCE_ID	(varchar 80). Identifier of the source.
PR_NAME	(varchar 80). Name of the processing rule.
PR_TYPE	(varchar 80). Type of processing rule.
PR_TEXT	(varchar 80). Text description of the processing rule.
PR_EFFECTIVE_TIME	(date). Starting date and time when the processing rule becomes effective. A value of 0 (zero). means it becomes effective immediately.
PR_EXPIRATION_TIME	(date). Date and time when the processing rule expires, or is no longer in effect. A value of 0 (zero). indicates the processing rule never expires.
PR_SUBTYPE	(varchar 80). Processing rule subtype.
RECORD_EXPIRED	(number 23). Date and time when this record becomes invalid.
RECORD_UPDATED	(char 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Message Processing Rules Match Criteria Table (MESSAGE_PR_MCH_T)

The Message Processing Rules Match Criteria table contains the criteria used to match messages with the instructions used to process them. When there is a match, specific processing rules are invoked that control how the message is to be processed.

Primary Keys

The MESSAGE_PR_MCH_T table contains the following primary keys:

- TRN_DATE
- TRN_NUMBER
- PR_SEQ_NUM
- PRMATCH_ID
- PRMATCH_COND
- TRN_TIMESTAMP

Column Descriptions

TRN_DATE (pk)	(date). Period date when the message came into being.
TRN_NUMBER (pk)	(number 8). Transaction reference sequence number. The first transaction message of each processing day is 00000001.
PR_SEQ_NUM (pk)	(number 10). Processing rule sequence number.
PRMATCH_ID (pk)	(varchar 40). Match ID.
PRMATCH_COND (pk)	(varchar 20). Match conditional.
PRMATCH_VALUE	(varchar 40). Not populated in this table.
TRN_TIMESTAMP (pk)	(number 23). Date and time when the record was written.
PRMATCH_LOGICAL	(varchar 10). Match logical.
PRMATCH_ACTION	(varchar 80). Description of the action to be taken when there is a match.
RECORD_EXPIRED	(number 23). Date and time when this record becomes invalid.

RECORD_UPDATED	(char 1). Indicates whether this row's timestamp column has been propagated to previous rows.
----------------	---

Message Processing Rules Parameters Table (MESSAGE_PR_PRM_T)

The Message Processing Rules Parameters table contains parameters that govern how and when particular processing rules are used to process messages.

Primary Keys

The MESSAGE_PR_PRM_T table contains the following primary keys:

- TRN_DATE
- TRN_NUMBER
- PR_SEQ_NUM
- PRPARAM_ID
- TRN_TIMESTAMP

Column Descriptions

TRN_DATE (pk)	(date). Period date when the message came into being.
TRN_NUMBER (pk)	(number 8). Transaction reference sequence number. The first transaction message of each processing day is 00000001.
PR_SEQ_NUM (pk)	(number 10). Processing rule sequence number.
PRPARAM_ID (pk)	(varchar 40). Processing rules parameter ID.
TRN_TIMESTAMP (pk)	(number 23). Date and time when the record was written.
PRPARAM_EDIT	(varchar 20). Edit type.
PRPARAM_COUNT	(number 10). Number of values in a single row.
RECORD_EXPIRED	(number 23). Date and time when this record becomes invalid.
RECORD_UPDATED	(char 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Processing Rules Parameter Values (MESSAGE_PR_PVL_T)

The Message Processing Rules Parameter Values table contains the values that match the parameters used in determining which processing rules are used to process a message.

Primary Keys

The MESSAGE_PR_PVL_T table contains the following primary keys:

- TRN_DATE
- TRN_NUMBER
- PR_SEQ_NUM
- PRPARAM_ID
- VALUE_SEQ_NUM
- TRN_TIMESTAMP

Column Descriptions

TRN_3DATE (pk)	(date). Period date when the message came into being.
TRN_NUMBER (pk)	(number 8). Transaction reference sequence number. The first transaction message of each processing day is 00000001.
PR_SEQ_NUM (pk)	(number 10). Processing rule sequence number.
PRPARAM_ID (pk)	(varchar 40). Processing rules parameter ID.
VALUE_SEQ_NUM (pk)	(number 10). Value sequence number.
TRN_TIMESTAMP (pk)	(number 23). Date and time when the record was written.
PARAMETER_VALUES	(varchar 80). Parameter values.
RECORD_EXPIRED	(number 23). Date and time when this record becomes invalid.
RECORD_UPDATED	(char 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Repetitive Tables

The Repetitive tables contain message templates for outgoing repetitive payments.

Parent/child relationships

The Repetitive table (REPETITIVE_T), Repetitive Credit table (REPETITIVE_CR_T), and the Repetitive Debit table (REPETITIVE_DR_T) combined make up one large repetitive record. This large table is split into three tables because of Oracle and DB2 column limits.

Primary keys

The Repetitive tables have the following primary keys in common:

- BANK
- TYPE
- KEY_ACC
- KEY_PAD
- REPETITIVE_ID
- RPT_TIME_STAMP

Repetitive Table (REPETITIVE_T)

The Repetitive table contains message templates for outgoing repetitive payments.

The repetitive templates contain information such as the account IDs and addresses of the parties involved in a transfer, the transaction type, and the method of payment.

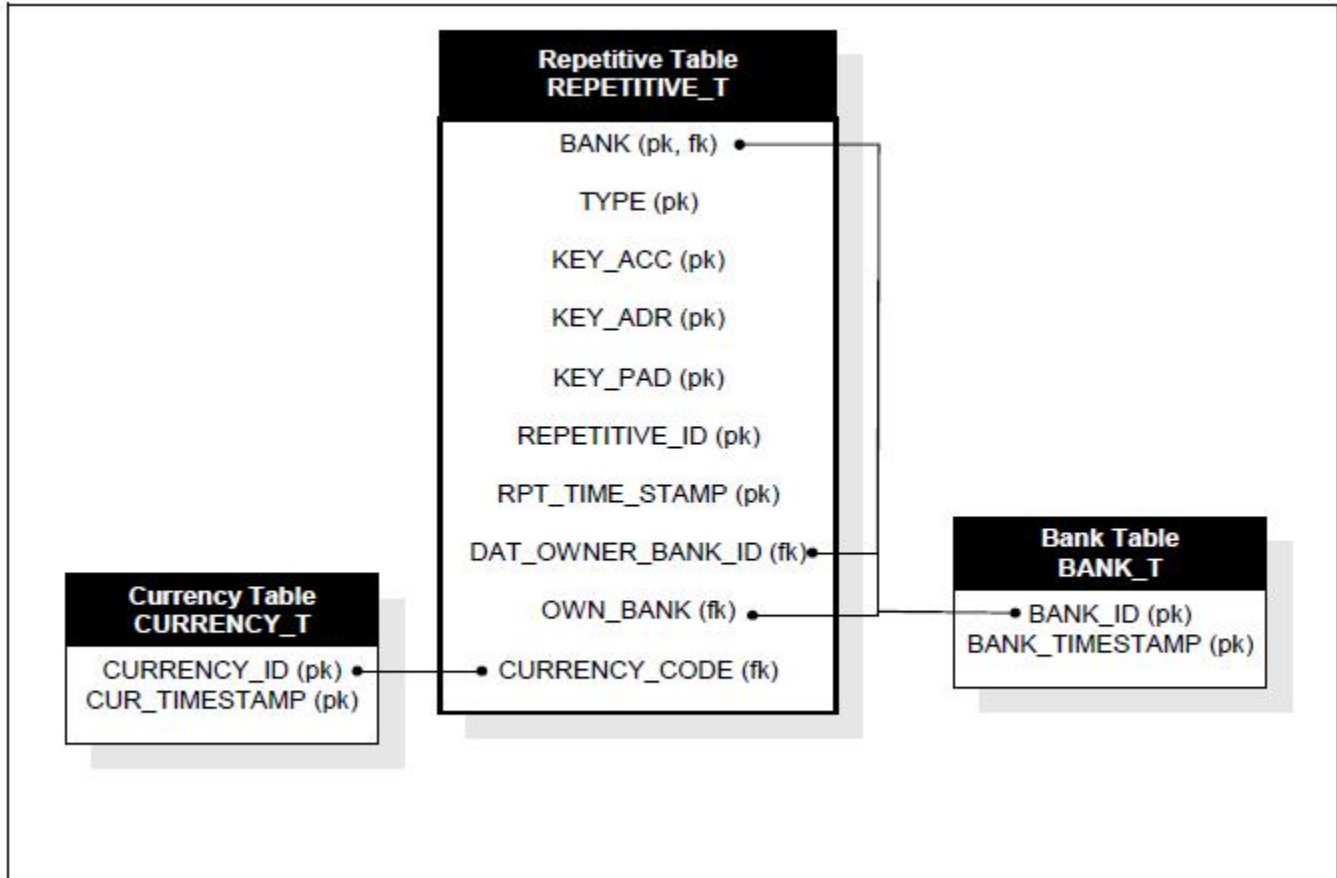
Table Relationships

The DAT_OWNER_BANK_ID column in the Repetitive table relates to the BANK_ID column in the Bank table.

The CURRENCY_CODE column in the Repetitive table relates to the CURRENCY_ID column in the Currency table.

Relationship Diagram

This relationship diagram shows the Repetitive table along with any tables that are related to it through foreign keys.



Column Descriptions

Many tables in the MTS database are modeled on a common data structure. However, not all columns in the data structure are used in every table. As a result, many columns exist in this table, but are not populated. These columns names are indicated by regular type (not bold).

BANK (pk, fk)	(varchar 3). Bank identifier of the bank that owns this repetitive record.
TYPE (pk)	(varchar 1). Account ID type; contains the following values: D (DDA) F (Foreign nostro account number) G (General ledger account number) P (CHIPS participant ID)

	V (Savings account number)
KEY_ACC (pk)	(varchar 30). Account number used to find an address.
KEY_ADR (pk)	(varchar 5). Address code used to distinguish among multiple addresses sharing a single account.
KEY_PAD (pk)	(varchar 1). Number of additional lines for addresses when a bank has several addresses.
REPETITIVE_ID (pk)	(varchar 8). Repetitive ID. Unique identifier assigned to the repetitive template that MTS uses to generate an outgoing transaction.
RPT_TIME_STAMP (pk)	(varchar 16). Date and time when the repetitive record was updated.
DAT_FUNCTION	(varchar 7). Function that last accessed the repetitive; contains the following values: VFY ADD (Verify add) VFY DEL (Verify delete) VFY UPD (Verify update)
CREATOR	(varchar 10). Operator ID and bank ID of the operator who created this repetitive record.
CREATE_DATE	(date). Date when this repetitive record was created.
UPDATER	(varchar 10). Operator ID and bank ID of the operator who updated this repetitive record.
UPDATE_DATE	(date). Date when this repetitive record was updated.
VERIFIER	(varchar 10). Operator ID and bank ID of the operator who verified this repetitive record.
VERIFY_DATE	(date). Date when this repetitive record was verified.
DAT_OWNER_PROD_ID	(varchar 3). Product ID associated with the repetitive template.
DAT_OWNER_BANK_ID (fk)	(varchar 3). Bank ID associated with the repetitive template.
DAT_OWNER_LOC	(varchar 6). Location ID associated with the repetitive template.
DAT_OWNER_CUST	(varchar 20). Customer ID associated with the repetitive template.
MSGTYPE	(varchar 9). Not populated in this table.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Message Tables

TRN_DATE	(date). Not populated in this table.
TRN_NUMBER	(varchar 8). Not populated in this table.
OWN_BANK (fk)	(varchar 3). Owning bank name.
OWN_LOC	(varchar 6). Owning bank location.
PROXY	(varchar 6). Not populated in this table.
SOURCE_CD	(varchar 3). Not populated in this table.
TYPE_CD	(varchar 4). Outgoing Federal Reserve, S.W.I.F.T., or CHIPS message type; correlates to the MSG_TYPE_TYP field in the MFE.
SUBTYPE	(varchar 4). Not populated in this table.
VALUE_DATE	(date). Not populated in this table.
TRAN_TYPE	(varchar 3). Transaction type. This column contains the following values: DEP (Deposit) DFA (Draft advice) DFT (Draft) DRR (Drawdown request) DRW (Drawdown response) FFB (Fed Funds bought) FFR (Fed Funds returned) FFS (Fed Funds sold) FTR (Funds transfer) NON (Nonaccounting)
FUNDS_TYPE	(varchar 1). Transaction type; mapped from the TYP field on the FTR screen. This column contains the following values: DEP (Deposit) DFA (Draft advice) DFT (Draft) DRR (Drawdown request) DRW (Drawdown response) FFB (Fed Funds bought) FFR (Fed Funds returned) FFS (Fed Funds sold) FTR (Funds transfer) NON (Nonaccounting)
INCOMING_MSGTYPE	(varchar 4). Not populated in this table.

INCOMING_FORMAT	(varchar 4). Not populated in this table.
INCOMING_REF	(varchar 80). Not populated in this table.
FRONTEND_REF_NUM	(varchar 16). Not populated in this table.
DOC_NUM	(varchar 7). Not populated in this table.
ITEM_NUM	(varchar 2). Not populated in this table.
FUNDS_CATEGORY	(varchar 10). Not populated in this table.
INSTR_ADV_TYPE	(varchar 3). Not populated in this table.
RECEIVE_CHAR	(number 10). Reserved for future use.
SEGMENT_INDEX	(number 10). Not populated in this table.
SPLIT_CTR	(number 5). Not populated in this table.
RPR_LEVEL	(varchar 2). Reserved for future use.
RECEIPT_TIME	(date). Not populated in this table.
INST_DATE	(date). Not populated in this table.
INST_TIME	(date). Not populated in this table.
POST_DATE	(date). Not populated in this table.
SEND_DATE	(date). Not populated in this table.
RELEASE_TIME_HH	(varchar 2). Not populated in this table.
RELEASE_TIME_MM	(varchar 2). Not populated in this table.
POST_TIME	(date). Not populated in this table.
AMOUNT	(number 21.3). Not populated in this table.
BASE_AMOUNT	(number 21.3). Not populated in this table.
CURRENCY_CODE (fk)	(varchar 3). Currency code specified in the repetitive template.
EXCH_RATE	(number 29.11). Not populated in this table.
FEDFUND_TYPE	(varchar 9). Not populated in this table.
FED_DAYS	(number 5). Not populated in this table.

Message Tables

INTEREST_RATE	(number 18.5). Not populated in this table.
TOTAL_AMOUNT	(number 21.3). Not populated in this table.
ANT_TOL_AMT	(number 21.3). Not populated in this table.
CUSIP (fk)	(varchar 9). Not populated in this table.
CUSIP_DESC	(varchar 20). Not populated in this table.
CAL	(varchar 4). Not populated in this table.
CAL_OVR	(varchar 1). Not populated in this table.
CALLER	(varchar 35). Not populated in this table.
PIN_REF_ID	(number 10). PIN reference ID.
VFC_PIN_REF_ID	(number 10). Not populated in this table.
PHN_CUS_BANK	(varchar 3). Bank identifier of the bank that owns this repetitive record.
PHN_CUS_TYPE	(varchar 1). Debit or credit party's account ID type. This column contains the following values: D (DDA) F (Foreign nostro account number) G (General ledger account number) P (CHIPS participant ID) V (Savings account number)
PHN_CUS_KEY_ACC	(varchar 30). Account identification number.
PHN_CUS_KEY_ADR	(varchar 5). Customer's address.
PHN_CUS_KEY_PAD	(varchar 1). Padding for account type.
TRADER_CTRL	(varchar 10). Not populated in this table.
TESTKEY_IN	(varchar 16). Not populated in this table.
TEST_TEXT	(varchar 24). Not populated in this table.
RISKINESS_IND	(varchar 2). Not populated in this table.
DBT_CHRG	(varchar 1). Flag indicating whether charges apply for debit. Values are bank-defined, determined from user-defined configuration routines for fee charges.

CDT_CHRG	(varchar 1). Credit analysis charge flag. Indicates whether charges apply for credit. Values are bank-defined, determined from user-defined configuration routines for fee charges.
COMMISSION	(varchar 1). Commission analysis charge flag. Indicates whether charges apply for commission. Values are bank-defined, determined from user-defined configuration routines for commission charges.
CBL_CHARGE	(varchar 1). Cable analysis charge flag. Indicates whether charges are to be applied for cable. Values are bank-defined, determined from user-defined configuration routines for cable charges.
COMM_MODE	(varchar 1). Commission charges.
CBL_MODE	(varchar 1). Cable charges.
DEST_COUNT	(number 5). Not populated in this table.
DEST_FUNC	(varchar 3). Not populated in this table.
DEST_Q_CLASS	(varchar 3). Not populated in this table.
DEST_OPR	(varchar 3). Not populated in this table.
ADM_LOCK_TEXT	(varchar 1). Not populated in this table.
BNK_TX	(varchar 1). Not populated in this table.
MULT_DBT	(varchar 1). Not populated in this table.
MULT_CDT	(varchar 1). Not populated in this table.
AMT_VFY	(varchar 1). Not populated in this table.
CUR_VFY	(varchar 1). Not populated in this table.
EXCH_VFY	(varchar 1). Not populated in this table.
SERIAL_VFY	(varchar 1). Not populated in this table.
CALL_NON_RPT	(varchar 1). Not populated in this table.
CALL_NOF	(varchar 1). Not populated in this table.
CALL_LIM	(varchar 1). Not populated in this table.
CALL_RPT	(varchar 1). Not populated in this table.
CALLBACK	(varchar 1). Not populated in this table.
FAIL_TST	(varchar 1). Not populated in this table.

Message Tables

RPTV_LIM	(varchar 1). Not populated in this table.
RPTV_CHNG	(varchar 1). Not populated in this table.
PDM	(varchar 1). Not populated in this table.
ADV_OVR	(varchar 1). Not populated in this table.
TEST_REQ	(varchar 1). Not populated in this table.
ACCT_VFC	(varchar 1). Not populated in this table.
CDT_ADV	(varchar 1). Not populated in this table.
MULTI_CUR	(varchar 1). Not populated in this table.
PRIORITY	(varchar 1). Not populated in this table.
DELIVERY	(varchar 1). Not populated in this table.
MON_INTERCEPT	(varchar 1). Not populated in this table.
NOTIFY	(varchar 1). Not populated in this table.
STOP_INTERCEPT	(varchar 1). Not populated in this table.
RTP_INTERCEPT	(varchar 1). Not populated in this table.
MATCHED_ANT	(varchar 1). Not populated in this table.
REPAIR_CHNG	(varchar 1). Not populated in this table.
CHIPS_RECVRY	(varchar 1). Not populated in this table.
NON_ACCTING	(varchar 1). Not populated in this table.
SWF_STMT	(varchar 1). Not populated in this table.
CHARGE_STATE	(varchar 1). Not populated in this table.
VFY_COUNT	(varchar 1). Not populated in this table.
MPA_DBT_AUTH	(varchar 1). Not populated in this table.
MPA_DBT_POST	(varchar 1). Not populated in this table.
MPA_CDT_AUTH	(varchar 1). Not populated in this table.
MPA_CDT_POST	(varchar 1). Not populated in this table.

TEST_STATE	(varchar 1). Not populated in this table.
STRAIGHT_THRU	(varchar 1). Not populated in this table.
PAY_STATE	(varchar 1). Not populated in this table.
EXCH_RATE_FLG	(varchar 1). Not populated in this table.
FX_TOLERANCE	(varchar 1). Not populated in this table.
VFC_COUNT	(varchar 1). Not populated in this table.
FUNDED_MSG	(varchar 1). Not populated in this table.
PMT_CHRG	(varchar 1). Not populated in this table.
INT_REROUTE	(varchar 1). Not populated in this table.
CUR_HOLD	(varchar 1). Not populated in this table.
OVR_OUT_LIM	(varchar 1). Not populated in this table.
FORCE_POST	(varchar 1). Not populated in this table.
RPTD_SWF_STMT	(varchar 1). Not populated in this table.
DBT_CNF_OVER	(varchar 1). Flag indicating whether the operator has overridden the debit confirmation (*) or not (blank) .
DBT_ADV_OVER	(varchar 1). Flag indicating whether the operator has overridden the debit advice (*) or not (blank) .
CDT_CNF_OVER	(varchar 1). Flag indicating whether the operator has overridden the credit confirmation (*) or not (blank) .
CDT_ADV_OVER	(varchar 1). Flag indicating whether the operator has overridden the credit advice (*) or not (blank) .
ALT_TO_RSK	(varchar 1). Flag indicating whether the payment is to be sent to risk if the RPT template is altered during payment entry (*) or not (blank) .
VFY_REQ	(varchar 1)Flag indicating whether the payment is to be routed as a nonrepetitive (*) or not (blank).
ALLOW_CHANGES	(varchar 1). Flag indicating whether the entry operator can make changes (Y). or not (N).
INTERNAL	(varchar 1). Flag indicating whether the repetitive was requested internally (Y) or not (N).
NOMODIFY	(varchar 1). Not populated in this table.

Message Tables

PRE_FOUND	(varchar 1). Not populated in this table.
FX_FOUND	(varchar 1). Not populated in this table.
CHECK_FOUND	(varchar 1). Not populated in this table.
COR_DONE	(varchar 1). Not populated in this table.
RTE_DONE	(varchar 1). Not populated in this table.
PMT_CHAR_CODE	(varchar 10). Payment characteristics code. Describes the nature of the payment based upon rules that are specific to the country's clearing system.
FIN_CPY_SRC_FLG	(varchar 1). R =MT102 request for CDT adv
REG_ID_TYPE	(varchar 1). Regulatory ID type.
REG_ID	(varchar 34). Regulatory ID.
BOPR_TEXT1	(varchar 35). Balance of payment reporting justification text line 1.
BOPR_TEXT2	(varchar 35). Balance of payment reporting justification text line 2.
BOPR_TEXT3	(varchar 35). Balance of payment reporting justification text line 3.
IMPOSED_CHAN	(varchar 1). Indicates whether a channel is imposed on the message that the repetitive creates. Y = channel is imposed
IMPOSED_AMOUNT	(varchar 1). Whether the repetitive has an amount imposed.

Repetitive Credit Table (REPETITIVE_CR_T)

The Repetitive Credit table provides credit information for repetitives.

Table Relationships

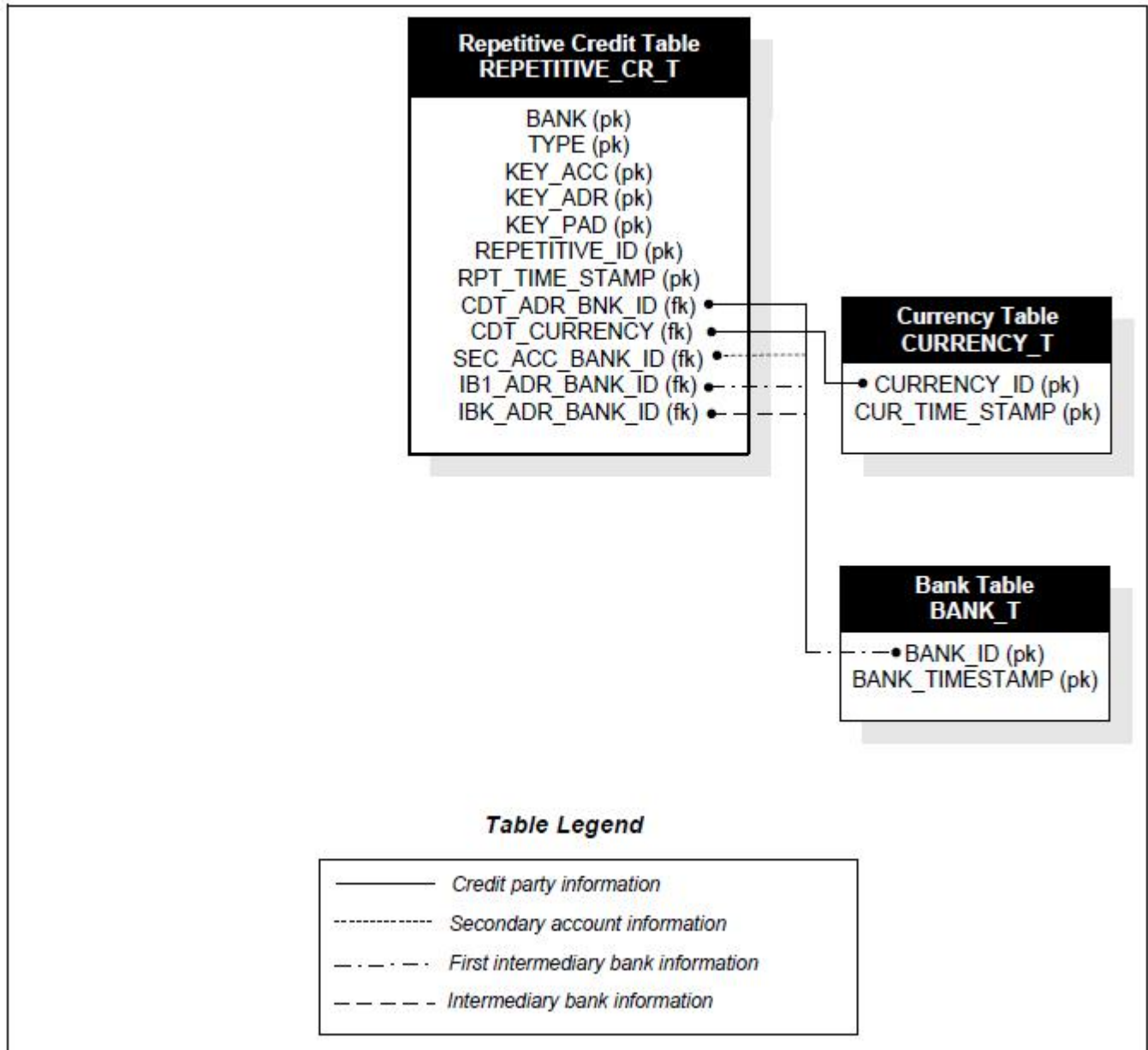
The following columns in the Repetitive Credit table relate to the BANK_ID column in the Bank table:

- CDT_ADR_BANK_ID
- SEC_ACC_BANK_ID
- IB1_ADR_BANK_ID
- IBK_ADR_BANK_ID

The CDT_CURRENCY column in the Repetitive Credit table relates to the CURRENCY_ID column in the Currency table.

Relationship Diagram

This relationship diagram shows the Repetitive Credit table along with the tables related to it through foreign keys.



Column Descriptions

Many tables in the MTS database are modeled on a common data structure. However, not all columns in the data structure are used in every table. As a result, many columns exist in this table, but are not populated. These columns names are indicated by regular type (not bold).

BANK (pk)	(varchar 3). Bank identifier of the bank that owns this repetitive record.
TYPE (pk)	(varchar 1). Account ID type; contains the following values: D (DDA) F (Foreign nostro account number) G (General ledger account number) P (CHIPS participant ID) V (Savings account number)
KEY_ACC (pk)	(varchar 30). Account number used to find an address.
KEY_ADR (pk)	(varchar 5). Address code used to distinguish among multiple addresses sharing a single account.
KEY_PAD (pk)	(varchar 1). Number of additional lines for addresses when a bank has several addresses.
REPETITIVE_ID (pk)	(varchar 8). Repetitive ID. Unique identifier assigned to the repetitive template that MTS uses to generate an outgoing transaction.
RPT_TIME_STAMP (pk)	(varchar 16). Date and time when the repetitive record was updated.
RECORD_EXPIRED	(varchar 16). Indicates the date and time at which this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
CDT_ADR_BNK_ID (fk)	(varchar 3). Credit party address bank ID.
CDT_REL_ID	(number 10). Unique system-assigned reference number for the credit party address.
CDT_OVR	(varchar 1). Credit party override flag; contains the following values: * (Credit party is not in REL). Space (Credit party is in REL). ? (There is an inconsistency in the REL database).
CDT_IDTYPE	(varchar 1). Credit party account ID type; mapped from the first part of CDT column on the FTR screen. This column contains the following values: A (Fedwire routing ABA number) B (branch code)

Message Tables

	C (CHIPS user ID) D (DDA) E (Extended ID) F (Foreign nostro account number) G (General ledger account number) I (Interbank lookup key) K (Customer ID number) M (Private leased line network MAC code) N (MTS mnemonic name) P (CHIPS participant ID) R (ACI Worldwide system address reference number) S (8- or 11-character S.W.I.F.T. address) T (Telex number) U (User-defined name)
	V (Savings account number) X (Telex answerback) Z (Cable address)
CDT_ID	(varchar 64). Credit party account ID.
BANK_ID	(varchar 3). Bank identifier of the bank that owns this repetitive record.
ID_TYPE	(vchar 1). Account ID type; contains the following values: D (DDA) F (Foreign nostro account number) G (General ledger account number) P (CHIPS participant ID) V (Savings account number)
ACCOUNT	(varchar 30). Account identification number.
CDT_AMOUNT	(number 21.3). Not populated in this table.
CDT_CURRENCY (fk)	(varchar 3). Currency of the credit party in the repetitive.
SEC_FLG	(varchar 16). Not populated in this table.
SEC_AMT	(number 21.3). Not populated in this table.
SEC_CUR	(varchar 3). Not populated in this table.
SEC_ACC_BANK_ID (fk)	(varchar 3). Bank identification number of secondary account.
SEC_ACC_ID_TYPE	(varchar 1). Secondary account ID type; contains the following values:

	D (DDA) F (Foreign nostro account number) G (General ledger account number) P (CHIPS participant ID) V (Savings account number)
SEC_ACC_ACCOUNT	(varchar 30). Secondary account number.
TER_FLG	(varchar 16). Not populated in this table.
TER_TER_AMT	(number 21.3). Not populated in this table.
TER_TER_CUR	(varchar 3). Not populated in this table.
TER_ACC_BANK_ID	(varchar 3). Not populated in this table.
TER_ACC_ID_TYPE	(varchar 1). Not populated in this table.
TER_ACC_ACCOUNT	(varchar 30). Not populated in this table.
CDT_ACC_CITY	(varchar 15). Not populated in this table.
CDT_ACC_CLASS	(varchar 6). Not populated in this table.
CDT_ACC_COUNTRY	(varchar 2). Not populated in this table.
CDT_ACC_PARENT	(varchar 10). Not populated in this table.
CDT_ACC_PRODS	(varchar 20). Not populated in this table.
CDT_SYS_OF_REC	(varchar 3). Not populated in this table.
CONCEN_BANK_ID	(varchar 3). Not populated in this table.
CONCEN_ID_TYPE	(varchar 1). Not populated in this table.
CONCEN_ACCOUNT	(varchar 30). Not populated in this table.
CDT_PARENT_CODE	(varchar 10). Not populated in this table.
CDT_ADR_CLASS	(varchar 6). Not populated in this table.
CDT_ADR_CITY	(varchar 15). Not populated in this table.
CDT_ADR_COUNTRY	(varchar 2). Not populated in this table.
CDT_ADR_TYPE	(varchar 6). Credit address type.
CDT_ADV_TYP	(varchar 3). Final payment advice type indicated by the sender for the credit party. This column contains the following values:

Message Tables

	CHK (Check) CHP (CHIPS) FED (Fedwire) LTR (No primary advice; secondary confirmation only) PHH (Phone hold) PHN (Phone) SWF (S.W.I.F.T.) TKT (Online ticket) WIR (Telex) Any user-defined INTRTL advice
CDT_ADR_ADV_TYPE	(varchar 3). Not populated in this table.
CDT_DEPARTMENT	(varchar 20). Bank-defined department to which the credit account is assigned.
CDT_LOCATION	(varchar 6). Not populated in this table.
CDT_RT_CODE	(varchar 6). Not populated in this table.
CDT_RT_STATE	(varchar 1). Not populated in this table.
CDT_SEC_SUB_ACCT	(varchar 61). Not populated in this table.
CDT_SHNAM	(varchar 20). Not populated in this table.
IDTYPE	(varchar 1). Not populated in this table.
IDKEY	(varchar 64). Not populated in this table.
ADV_TYPE	(varchar 3). Not populated in this table.
PHN_TIME	(varchar 4). Not populated in this table.
PYMNT_Q_PROD_ID	(varchar 3). Product ID.
PYMNT_Q_BANK_ID	(varchar 3). Bank ID.
PYMNT_Q_LOC	(varchar 6). Not populated in this table.
PYMNT_Q_CUST	(varchar 20). Customer ID.
PYMNT_Q_NAME	(varchar 33). Queue name.
CDT_NAME1	(varchar 35). First line of the credit party's name.
CDT_NAME2	(varchar 35). Second line of the credit party's name.
CDT_NAME3	(varchar 35). Third line of the credit party's name.

CDT_NAME4	(varchar 35). Fourth line of the credit party's name.
CDT_ADV_INST1	(varchar 35). First line of the credit advice instruction.
CDT_ADV_INST2	(varchar 35). Second line of the credit advice instruction.
CDT_ADV_INST3	(varchar 35). Third line of the credit advice instruction.
CDT_RECON_REF	(varchar 12). Credit reconciliation reference.
CDT_SPC_INST1	(varchar 35). First line of the credit special instructions.
CDT_SPC_INST2	(varchar 35). Second line of the credit special instructions.
CDT_SPC_INST3	(varchar 35). Third line of the credit special instructions.
CDT_HOLD_FLG	(varchar 1). Not populated in this table.
CDT_PAY_FLG	(varchar 1). Not populated in this table.
CDT_SECWIR_FLG	(varchar 1). Not populated in this table.
CDT_ALT_SWIFT_FLG	(varchar 1). Not populated in this table.
CDT_NO_PHNADV_FLG	(varchar 1). Not populated in this table.
CDT_DELIVERY_FLG	(varchar 1). Not populated in this table.
CDT_NOF_LOOKED_UP	(varchar 1). Reserved for future use.
CDT_ID_CHG_FLG	(varchar 1). Not populated in this table.
IB1_ADR_BNK_ID (fk)	(varchar 3). First intermediary bank party address identification number.
IB1_REL_ID	(number 10). Unique system-generated reference number for the first intermediary bank party address.
IB1_SECWIR	(varchar 1). Flag indicating whether the first intermediary bank is sent a secondary wire (Y). or not (N).
IB1_IDTYPE	(varchar 1). First intermediary bank party ID type.
IB1_ID	(varchar 64). First intermediary bank ID type. This column contains the following values: Space (Foreign account) C (CHIPS user) E (Extended ID) S (S.W.I.F.T. ID) X (Telex answer-back)

Message Tables

IB1_NAME1	(varchar 35). First line of the first intermediary bank name and address.
IB1_NAME2	(varchar 35). Second line of the first intermediary bank name and address.
IB1_NAME3	(varchar 35). Third line of the first intermediary bank name and address.
IB1_NAME4	(varchar 35). Fourth line of the first intermediary bank name and address.
IB1_ADV_INST1	(varchar 35). First line of additional text to send to the first intermediary bank.
IB1_ADV_INST2	(varchar 35). Second line of additional text to send to the first intermediary bank.
IB1_ADV_INST3	(varchar 35). Third line of additional text to send to the first intermediary bank.
IBK_ADR_BNK_ID (fk)	(varchar 3). Intermediary bank party address identification number.
IBK_REL_ID	(number 10). Unique system-generated reference number for the intermediary bank party address.
IBK_SECWIR	(varchar 1). Flag indicating whether the intermediary bank is sent a secondary wire (Y) or not (N).
IBK_IDTYPE	(varchar 1). Intermediary bank ID type. This column contains the following values: Space (Foreign account) C (CHIPS user) E (Extended ID) S (S.W.I.F.T. ID) X (Telex answer-back)
IBK_ID	(varchar 64). Intermediary bank identifier. The first intermediary bank can be referenced by multiple IDs.
IBK_NAME1	(varchar 35). First line of intermediary bank name and address.
IBK_NAME2	(varchar 35). Second line of intermediary bank name and address.
IBK_NAME3	(varchar 35). Third line of intermediary bank name and address.
IBK_NAME4	(varchar 35). Fourth line of intermediary bank name and address.
IBK_ADV_INST1	(varchar 35). First line of additional text to send to the intermediary bank.
IBK_ADV_INST2	(varchar 35). Second line of additional text to send to the intermediary bank.
IBK_ADV_INST3	(varchar 35). Third line of additional text to send to the intermediary bank.
BBK_ADR_BNK_ID	(varchar 3). Beneficiary bank party address identification number.

BBK_REL_ID	(number 10). Unique system-generated reference number for the beneficiary bank party address.
BBKF_SECWIR	(varchar 1). Flag indicating whether the beneficiary bank is sent a secondary wire (Y) or not (N).
BBKF_IDTYPE	(varchar 1). Beneficiary bank party ID type. This column contains the following values: Space (Foreign account) C (CHIPS user) E (Extended ID) S (S.W.I.F.T. ID) X (Telex answer-back)
BBKF_ID	(varchar 64). Beneficiary bank party ID. The beneficiary bank can be referenced by multiple IDs.
BBK_NAME1	(varchar 35). First line of the beneficiary bank account name and address.
BBK_NAME2	(varchar 35). Second line of the beneficiary bank account name and address.
BBK_NAME3	(varchar 35). Third line of the beneficiary bank account name and address.
BBK_NAME4	(varchar 35). Fourth line of the beneficiary bank account name and address.
BBK_ADV_INST1	(varchar 35). First line of additional text to send to the beneficiary bank.
BBK_ADV_INST2	(varchar 35). Second line of additional text to send to the beneficiary bank.
BBK_ADV_INST3	(varchar 35). Third line of additional text to send to the beneficiary bank.
BNP_CHARGE_FLG	(varchar 1). Flag indicating which party to charge; mapped from the CH column on the FTR screen. This column contains the following values: B (Beneficiary). or O (Originator).
BNP_IDTYPE	(varchar 1). Beneficiary ID type; contains the following values: Space (Foreign account) C (CHIPS user) E (Extended ID) S (S.W.I.F.T. ID) X (Telex answer-back)
BNP_ID	(varchar 64). Beneficiary party ID. The beneficiary can be referenced by multiple IDs.
BNP_BNK_FLG	(varchar 1). Flag indicating whether beneficiary party is a bank (Y) or not (N).
BNP_NAME1	(varchar 35). First line of the beneficiary account name and address.

Message Tables

BNP_NAME2	(varchar 35). Second line of the beneficiary account name and address.
BNP_NAME3	(varchar 35). Third line of the beneficiary account name and address.
BNP_NAME4	(varchar 35). Fourth line of the beneficiary account name and address.
BNP_ADV_INST1	(varchar 35). First line of additional text to send to the beneficiary bank.
BNP_ADV_INST2	(varchar 35). Second line of additional text to send to the beneficiary bank.
BNP_ADV_INST3	(varchar 35). Third line of additional text to send to the beneficiary bank.
ORP_BEN_INF1	(varchar 35). First line of additional text to send to the beneficiary party from the originating party.
ORP_BEN_INF2	(varchar 35). Second line of additional text to send to the beneficiary party from the originating party.
ORP_BEN_INF3	(varchar 35). Third line of additional text to send to the beneficiary party from the originating party.
ORP_BEN_INF4	(varchar 35). Fourth line of additional text to be sent to the beneficiary party from the originating party.
CDT_PAYSYS_FMT	(varchar 1). Indicates whether a payment has been formatted.
IB1_CHIPS_UPTO	(number 10). First intermediary bank's length of the part not from the CHIPS UID database.
IB1_CHIPS_QUAL	(number 10). First intermediary bank's length of the part from the CHIPS UID database.
IBK_CHIPS_UPTO	(number 10). Intermediary bank's length of the part not from the CHIPS UID database.
IBK_CHIPS_QUAL	(number 10). Intermediary bank's length of the part from the CHIPS UID database.
BBK_CHIPS_UPTO	(number 10). Beneficiary bank's length of the part not from the CHIPS UID database.
BBK_CHIPS_QUAL	(number 10). Beneficiary bank's length of the part from the CHIPS UID database.
BNP_CHIPS_UPTO	(number 10). Beneficiary's length of the part not from the CHIPS UID database.
BNP_CHIPS_QUAL	(number 10). Beneficiary's length of the part from the CHIPS UID database.
CDT_STATE	(varchar 3). State or province of the credit party.
CDT_POSTAL_CODE	(varchar 15). Postal code of the credit party.
CDT_RES_COUNTRY	(varchar 2). Resident country of the credit party.

CDT_REF_NUMBER	(varchar 16). Reference number of the credit party.
IBK_STATE	(varchar 3). State or province of the intermediary bank party.
IBK_POSTAL_CODE	(varchar 15). Postal code of the intermediary bank party.
IBK_RES_COUNTRY	(varchar 2). Resident country of the intermediary bank party.
IBK_REF_NUM	(varchar 16). Reference number of the intermediary bank party.
IB1_STATE	(varchar 3). State or province of the first intermediary bank party.
IB1_POSTAL_CODE	(varchar 15). Postal code of the first intermediary bank party.
IB1_RES_COUNTRY	(varchar 2). Resident country of the first intermediary bank party.
IB1_REF_NUM	(varchar 16). Reference number of the first intermediary bank party.
BBK_STATE	(varchar 3). State or province of the beneficiary bank party.
BBK_POSTAL_CODE	(varchar 15). Postal code of the beneficiary bank party.
BBK_RES_COUNTRY	(varchar 2). Resident country of the beneficiary bank party.
BBK_REF_NUM	(varchar 16). Reference number of the beneficiary bank party.
BNP_STATE	(varchar 3). State or province of the beneficiary party.
BNP_POSTAL_CODE	(varchar 15). Postal code of the beneficiary party.
BNP_RES_COUNTRY	(varchar 2). Resident country of the beneficiary party.
BNP_REF_NUM	(varchar 16). Reference number of the beneficiary party.
BNP_MAILING_COUNTRY	(varchar 2). Mailing country for the beneficiary party.
REGULATORY_REPORT1	(varchar 35). First field of the regulatory report (77B).
REGULATORY_REPORT2	(varchar 35). Second field of the regulatory report (77B).
REGULATORY_REPORT3	(varchar 35). Third field of the regulatory report (77B).
BNP_ADR_BNK_ID	(varchar 3). Beneficiary party address bank ID.
BNP_REL_ID	(varchar 11). Beneficiary party address reference ID.
CDT_CAMEFROM	(varchar 20). Source where the credit party was found.
IB1_CAMEFROM	(varchar 20). Source where the first intermediary bank party was found.

Message Tables

IBK_CAMEFROM	(varchar 20). Source where the intermediary bank party was found.
BBK_CAMEFROM	(varchar 20). Source where the beneficiary bank party was found.
BNP_CAMEFROM	(varchar 20). Source where the beneficiary party was found.
CREDITSIDE_RESIDENCY	(varchar 2). Residency of the ultimate credit party, for example, the originating party, ordering bank, or credit party.
CDT_PROFILE_IDBANK	(varchar 3). Bank for the credit party profile.
CDT_PROFILE_IDTYPE	(varchar 1). ID type for the credit party profile.
CDT_PROFILE_IDACC	(varchar 30). Account number for the credit party profile.
CDT_PROFILE_IDADR	(varchar 5). Address code for the credit party profile.
CDT_PROFILE_IDPAD	(varchar 1). Padding for the credit party profile.
CDT_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the credit party.
IBK_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the intermediary bank party.
IB1_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the first intermediary bank party.
BBK_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the beneficiary party.
BNP_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the beneficiary party.

Repetitive Debit Table (REPETITIVE_DR_T)

The Repetitive Debit table provides debit information for repetitives.

Table Relationships

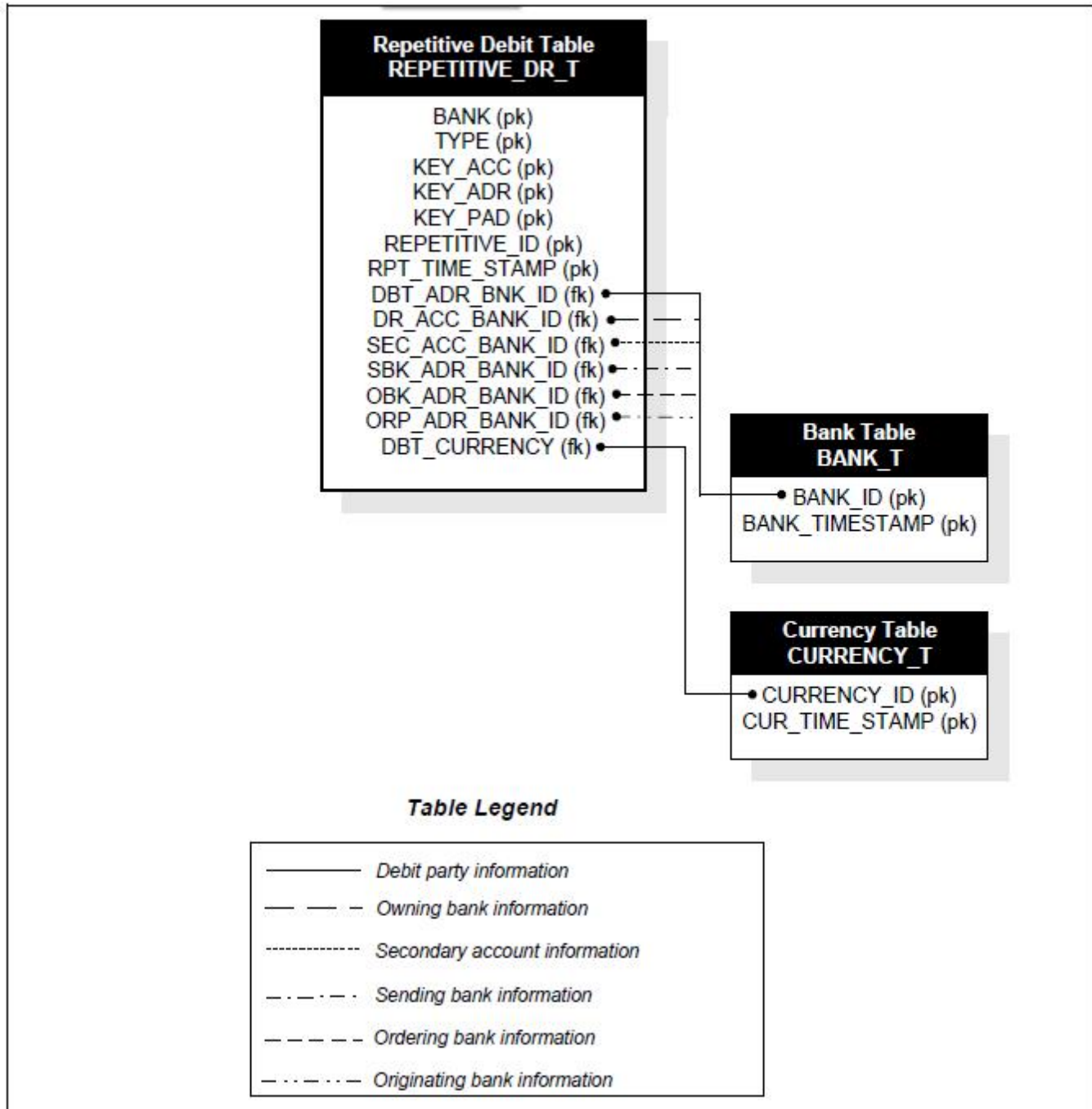
The following columns in the Repetitive Debit table relate to the BANK_ID column in the Bank table:

- DBT_ADR_BANK_ID
- DR_ACC_BANK_ID
- SEC_ACC_BANK_ID
- SBK_ADR_BANK_ID
- OBK_ADR_BANK_ID
- ORP_ADR_BANK_ID

The DBT_CURRENCY column in the Repetitive Debit table relates to the CURRENCY_ID column in the Currency table.

Relationship Diagram

This relationship diagram shows the Repetitive Debit table along with the tables related to it through foreign keys.



Column Descriptions

Many tables in the MTS database are modeled on a common data structure. However, not all columns in the data structure are used in every table. As a result, many columns exist in this table, but are not populated. These columns names are indicated by regular type (not bold).

BANK (pk)	(varchar 3). Bank identifier of the bank that owns this repetitive record.
TYPE (pk)	(varchar 1). Account ID type; contains the following values: D (DDA) F (Foreign nostro account number) G (General ledger account number) P (CHIPS participant ID) V (Savings account number)
KEY_ACC (pk)	(varchar 30). Account number used to find an address.
KEY_ADR (pk)	(varchar 5). Address code used to distinguish among multiple addresses sharing a single account.
KEY_PAD (pk)	(varchar 1). Number of additional lines for addresses when a bank has several addresses.
REPETITIVE_ID (pk)	(varchar 8). Repetitive ID. Unique identifier assigned to the repetitive template that MTS uses to generate an outgoing transaction.
RPT_TIME_STAMP (pk)	(varchar 16). Date and time when the repetitive record was updated.
RECORD_EXPIRED	(varchar 16). Date and time at which the record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
DBT_ADR_BNK_ID (fk)	(varchar 3). Debit party address bank identification number.
DBT_REL_ID	(number 10). Unique system-generated reference number for this debit party address.
DBT_TYP_OVR	(varchar 1). Debit party override flag; contains the following values: * (Debit party is not in REL) Space (Debit party is in REL) ? (There is an inconsistency in the REL database)
DBT_TYP_IDTYPE	(varchar 1). Debit party account ID type; mapped from the first part of DBT column on the FTR screen. This column contains the following values: A (Fedwire routing ABA number) B (Branch code)

Message Tables

	C (CHIPS user ID) D (DDA) E (Extended ID) F (Foreign nostro account number) G (General ledger account number) I (Interbank lookup key) K (Customer ID number) M (Private leased line network MAC code) N (MTS mnemonic name) P (CHIPS participant ID) R (ACI Worldwide system address reference number) S (8- or 11-character S.W.I.F.T. address) T (Telex number) U (User-defined name)
	V (Savings account number) X (Telex answerback) Z (Cable address)
DBT_TYP_ID	(varchar 64). Debit party account ID.
DR_ACC_BANK_ID (fk)	(varchar 3). Bank identifier of the bank that owns this repetitive record.
DR_ACC_ID_TYPE	(vchar 1). Account ID type; contains the following values: D (DDA) F (Foreign nostro account number) G (General ledger account number) P (CHIPS participant ID) V (Savings account number)
DR_ACC_ACCOUNT	(varchar 30). Account identification number.
DBT_AMOUNT	(number 21.3). Not populated in this table.
DBT_CURRENCY (fk)	(varchar 3). Currency of the debit party in the repetitive.
SEC_FLG	(varchar 16). Not populated in this table.
SEC_AMT	(number 21.3). Not populated in this table.
SEC_CUR	(varchar 3). Not populated in this table.
SEC_ACC_BANK_ID (fk)	(varchar 3). Bank identification number of secondary account.
SEC_ACC_ID_TYPE	(varchar 1). Secondary account ID type; contains the following values:

	D (DDA) F (Foreign nostro account number) G (General ledger account number) P (CHIPS participant ID) V (Savings account number)
SEC_ACC_ACCOUNT	(varchar 30). Secondary account number.
TER_FLG	(varchar 16). Not populated in this table.
TER_AMT	(number 21.3). Not populated in this table.
TER_CUR	(varchar 3). Not populated in this table.
TER_ACC_BANK_ID	(varchar 3). Not populated in this table.
TER_ACC_ID_TYPE	(varchar 1). Not populated in this table.
TER_ACC_ACCOUNT	(varchar 30). Not populated in this table.
DBT_ACC_CLASS	(varchar 6). Not populated in this table.
DBT_ACC_CITY	(varchar 15). Not populated in this table.
DBT_ACC_COUNTRY	(varchar 2). Not populated in this table.
DBT_ACC_PARENT	(varchar 10). Not populated in this table.
DBT_ACC_PRODS	(varchar 20). Not populated in this table.
DBT_DEPARTMENT	(varchar 20). Bank-defined department where the debit account is assigned.
DBT_FEE_CODE	(varchar 8). Not populated in this table.
DBT_SEC_SUB_ACCT	(varchar 61). Not populated in this table.
DBT_SYS_OF_REC	(varchar 3). Not populated in this table.
CONCEN_BANK_ID	(varchar 3). Not populated in this table.
CONCEN_ID_TYPE	(char 1). Not populated in this table.
CONCEN_ACCOUNT	(varchar 30). Not populated in this table.
DBT_PARENT_CODE	(varchar 10). Not populated in this table.
DBT_ADR_TYPE	(varchar 6). Not populated in this table.
DBT_ADR_CLASS	(varchar 6). Not populated in this table.

Message Tables

DBT_ADR_CITY	(varchar 15). Not populated in this table.
DBT_ADR_COUNTRY	(varchar 2). Not populated in this table.
DBT_ADV_TYP	(varchar 3). Not populated in this table.
DBT_RT_CODE	(varchar 6). Not populated in this table.
DBT_RT_STATE	(varchar 1). Not populated in this table.
DR_WIR_IDTYPE	(varchar 1). Not populated in this table.
DR_WIR_IDKEY	(varchar 64). Not populated in this table.
DBT_NAME1	(varchar 35). First line of the debit party's name.
DBT_NAME_2	(varchar 35). Second line of the debit party's name.
DBT_NAME_3	(varchar 35). Third line of the debit party's name.
DBT_NAME_4	(varchar 35). Fourth line of the debit party's name.
DBT_DRAFT_NUM	(varchar 16). Not populated in this table.
DBT_RECON_REF	(varchar 12). General ledger reconciliation reference number.
DBT_SPC_INST1	(varchar 35). First line of the debit special instructions.
DBT_SPC_INST2	(varchar 35). Second line of the debit special instructions.
DBT_SPC_INST3	(varchar 35). Third line of the debit special instructions.
DBT_HOLD	(varchar 1). Not populated in this table.
DBT_BAL	(varchar 1). Not populated in this table.
DBT_LIM	(varchar 1). Not populated in this table.
CHIPS_LIM	(varchar 1). Not populated in this table.
DBT_GRP_BAL	(varchar 1). Not populated in this table.
DBT_DRAWDOWN	(varchar 1). Not populated in this table.
DBT_PRE	(varchar 1). Not populated in this table.
DBT_GRP_PRE	(varchar 1). Not populated in this table.
DBT_PS_ELIG	(varchar 1). Flag indicating whether the debit party is eligible for pre or split advising. This column contains the following values:

	<p>S (Eligible for split-date accounting)</p> <p>P (Eligible for preadvicing)</p> <p>N (Eligible for send-date accounting, not preadvicing)</p> <p>L (Not eligible; preadvise limit not set, but required)</p> <p>X (Not eligible; preadvise limit has expired)</p> <p>Blank (Not relevant)</p>
DBT_ITEM_HOLD	(varchar 1). Not populated in this table.
DBT_NOF_LOOKED_UP	(varchar 1). Reserved for future use.
DDA_BALANCE	(varchar 9). Not populated in this table.
BALANCE_RISK	(varchar 8). Not populated in this table.
SBK_ADR_BNK_ID (fk)	(varchar 3). Sending bank's address identification number.
SBK_REL_ID	(number 10). Unique system-generated reference number for the sending bank party address.
SBK_IDTYPE	<p>(varchar 1). Sending bank identification type. This column contains the following values:</p> <p>Space (Foreign account)</p> <p>C (CHIPS user)</p> <p>E (Extended identification number)</p> <p>S (S.W.I.F.T. identification number)</p> <p>X (Telex answer-back)</p>
SBK_ID	(varchar 64). Sending bank identification number.
NAME1	(varchar 35). First line of the sending bank name.
NAME2	(varchar 35). Second line of the sending bank name.
NAME3	(varchar 35). Third line of the sending bank name.
NAME4	(varchar 35). Fourth line of the sending bank name.
REF_NUM	(varchar 16). Sending bank reference number.
OBK_ADR_BNK_ID (fk)	(varchar 3). Ordering bank address identification number.
OBK_REL_ID	(number 10). Unique system-generated reference number for the ordering bank party address.
OBK_IDTYPE	<p>(varchar 1). Ordering bank identification type. This column contains the following values:</p> <p>Space (Foreign account)</p>

Message Tables

	C (CHIPS user) E (Extended identification number) S (S.W.I.F.T. identification number) X (Telex answer-back)
OBK_ID	(varchar 64). Ordering bank identification number.
OBK_NAME1	(varchar 35). First line of the ordering bank name.
OBK_NAME2	(varchar 35). Second line of the ordering bank name.
OBK_NAME3	(varchar 35). Third line of the ordering bank name.
OBK_NAME4	(varchar 35). Fourth line of the ordering bank name.
OBK_REF_NUM	(varchar 16). Ordering bank reference number.
ORP_ADR_BNK_ID (fk)	(varchar 3). Originator bank address identification number.
ORP_REL_ID	(number 10). Unique system-generated reference number for the originating bank party address.
ORP_IDTYPE	(vchar 1). Originator identification type. This column contains the following values: Space (Foreign account) C (CHIPS user) E (Extended identification number) S (S.W.I.F.T. identification number) X (Telex answer-back)
ORP_ID	(varchar 64). Originator identification number.
ORP_NAME1	(varchar 35). First line of the originator name.
ORP_NAME2	(varchar 35). Second line of the originator name.
ORP_NAME3	(varchar 35). Third line of the originator name.
ORP_NAME4	(varchar 35). Fourth line of the originator name.
ORP_REF_NUM	(varchar 16). Originator reference number.
DBT_BNK_INF1	(varchar 35). Not populated in this table.
DBT_BNK_INF2	(varchar 35). Not populated in this table.
DBT_BNK_INF3	(varchar 35). Not populated in this table.
DBT_BNK_INF4	(varchar 35). Not populated in this table.

DBT_BNK_INF5	(varchar 35). Not populated in this table.
DBT_BNK_INF6	(varchar 35). Not populated in this table.
INS_ADR_BNK_ID	(varchar 3). Instructing bank ID.
INS_REL_ID	(number 10). Instructing bank REL ID.
INS_IDTYPE	(varchar 1). Instructing bank ID type.
INS_ID	(varchar 64). Instructing bank ID.
INS_NAME1	(varchar 35). Instructing bank name, line 1.
INS_NAME2	(varchar 35). Instructing bank name, line 2.
INS_NAME3	(varchar 35). Instructing bank name, line 3.
INS_NAME4	(varchar 35). Instructing bank name, line 4.
INS_REF_NUM	(varchar 16). Instructing bank reference number.
DBT_STATE	(varchar 3). State or province of the debit party.
DBT_POSTAL_CODE	(varchar 15). Postal code of the debit party.
DBT_RES_COUNTRY	(varchar 2). Resident country of the debit party.
DBT_REF_NUM	(varchar 16). Reference number of the debit party.
SBK_STATE	(varchar 3). State or province of the sending bank party.
SBK_POSTAL_CODE	(varchar 15). Postal code of the sending bank party.
SBK_RES_COUNTRY	(varchar 2). Resident country of the sending bank party.
OBK_STATE	(varchar 3). State or province of the ordering bank party.
OBK_POSTAL_CODE	(varchar 15). Postal code of the ordering bank party.
OBK_RES_COUNTRY	(varchar 2). Resident country of the ordering bank party.
ORP_STATE	(varchar 3). State province of the originator party.
ORP_POSTAL_CODE	(varchar 15). Postal code of the originator party.
OPR_RES_COUNTRY	(varchar 2). Resident country of the originator party.
INS_STATE	(varchar 3). State or province of the instructing bank party.

Message Tables

INS_POSTAL_CODE	(varchar 15). Postal code of the instructing bank party.
INS_RES_COUNTRY	(varchar 2). Resident country of the instructing bank party.
RCA_STATE	(varchar 3). State or province of the receiving agent.
RCA_POSTAL_CODE	(varchar 15). Postal code of the receiving agent.
RCA_RES_COUNTRY	(varchar 2). Resident country of the receiving agent.
DBT_CAMEFROM	(varchar 20). Source where the debit party was found.
SBK_CAMEFROM	(varchar 20). Source where the sending bank party was found.
OBK_CAMEFROM	(varchar 20). Source where the ordering bank party was found.
ORP_CAMEFROM	(varchar 20). Source where the originator party was found.
INS_CAMEFROM	(varchar 20). Source where the instructing party was found.
RCA_CAMEFROM	(varchar 20). Source where the receiving agent party was found.
DEBITSIDE_RESIDENCY	(varchar 2). Residency of the ultimate debit party, for example, the originating party, ordering bank, or debit party.
DBT_PROFILE_IDBANK	(varchar 3). Bank for the debit party profile.
DBT_PROFILE_IDTYPE	(varchar 1). ID type for the debit party profile.
DBT_PROFILE_IDACC	(varchar 30). Account number for the debit party profile.
DBT_PROFILE_IDADR	(varchar 5). Address code for the debit party profile.
DBT_PROFILE_IDPAD	(varchar 1). Padding for the debit party profile.
SBK_PROFILE_IDBANK	(varchar 3). Bank for the sending bank party profile.
SBK_PROFILE_IDTYPE	(varchar 1). ID type for the sending bank party profile.
SBK_PROFILE_IDACC	(varchar 30). Account number for the sending bank profile.
SBK_PROFILE_IDADR	(varchar 5). Address code for the sending bank profile.
SBK_PROFILE_IDPAD	(varchar 1). Padding for the sending bank profile.
DBT_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the debit party.
SBK_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the sending bank party.
OBK_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the ordering bank party.

ORP_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the originator party.
INS_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the instructing bank party.
RCA_BEI_FLAG	(varchar 1). SWIFT BIC/BEI flag of the receiving agent party.

Auxiliary Tables

MTS can be set up to use data from external sources, for example, to retrieve information needed to send a money transfer or to provide a lookup capability when you need information that is not in your own Relationship File.

The AUX tables contain the same type of information as is found in the REL database, but much less of it. The data may include lists of ABA routing numbers, CHIPS Universal IDs (UIDs), S.W.I.F.T. Bank Identifier Codes (BICs), and bank DDA and savings account numbers. It's primary function is to store information in addition to what is in REL, rather than duplicate REL.

Parent/child relationships

The Auxiliary Address table (AUX_ADDRESS_T) is the parent table to the Auxiliary ABA DDA table (AUX_ABA_DDA_T) and the Auxiliary Notes table (AUX_NOTES_T).

Primary keys

The auxiliary tables have these primary keys in common:

- TYPE
- KEY_ACC
- AUX_TIME_STAMP

Auxiliary Address Table (AUX_ADDRESS_T)

The Auxiliary Address table provides information pertaining to banks associated with a network.

For each bank on the network, the table provides the identifier code pertaining to the network, (for example, a SWIFT BIC, CHIPS UID, or an ABA routing number), as well as the bank's address and other related information.

Bank table

The following columns in the Auxiliary Address table relate to the BANK_ID column in the Bank table:

- BANK
- DAT_OWNER_BANK_ID

Account table

The following table lists the foreign keys in the Auxiliary Address table that relate to columns in the Account table:

These Auxiliary Address table columns	Relate to these Account table columns...
CONCEN_BANK_ID CONCEN_ID_TYPE CONCEN_ACCOUNT	BANK_ID ID_TYPE ACCOUNT

Algorithm table

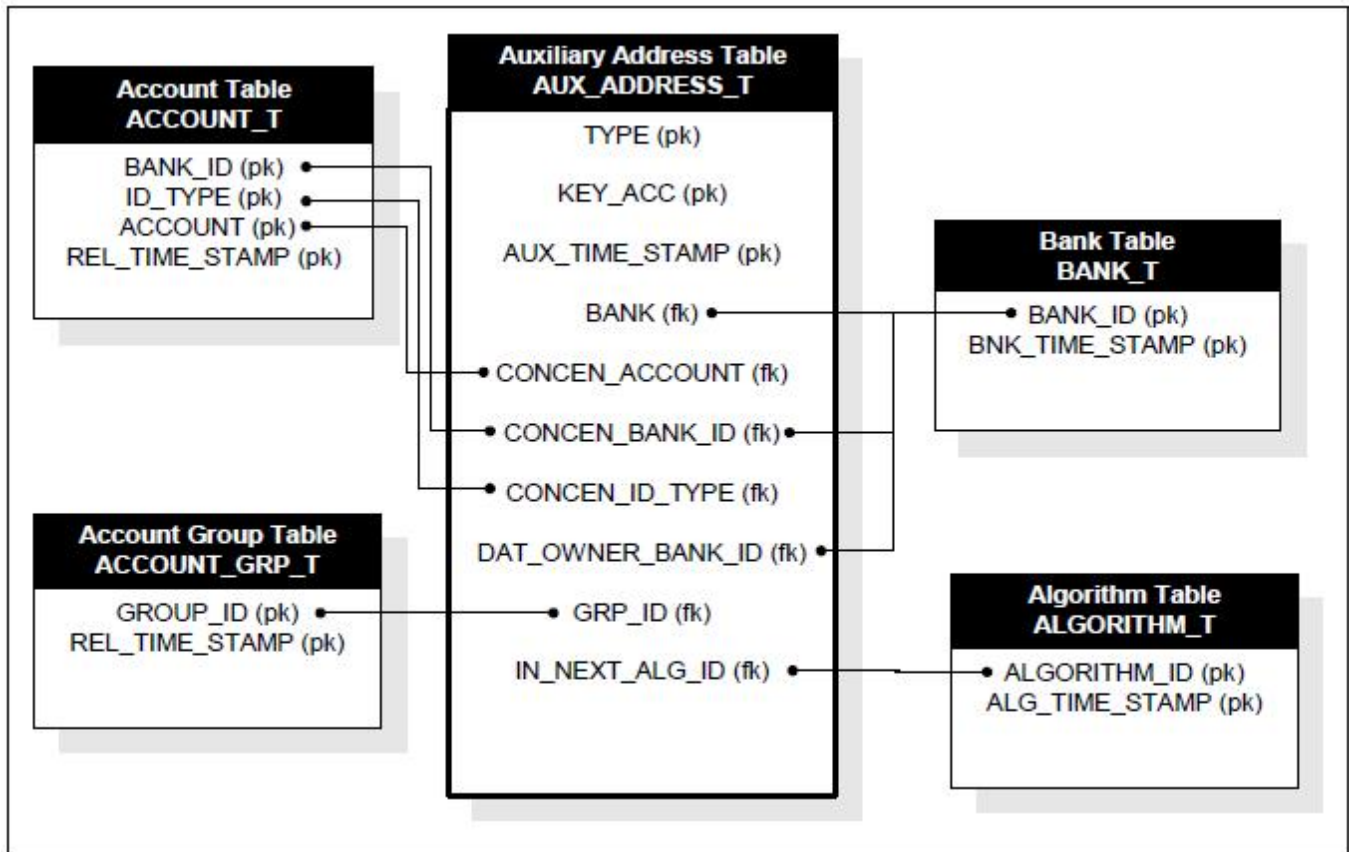
The IN_NEXT_ALG_ID column in the Auxiliary Address table relates to the ALGORITHM_ID column in the Algorithm table.

Account Group table

The GRP_ID column in the Auxiliary Address table relates to the GROUP_ID column in the Account Group table.

Relationship Diagram

This relationship diagram shows the Auxiliary Address table along with any tables that are related to it through foreign keys.



Column Descriptions

Several columns in this table are not used because addresses in an auxiliary file do not have accounts associated with them.

TYPE (pk)	(varchar 1). Address ID type. This column is part of a local, bank-defined customer information file. This column contains the following values: <ul style="list-style-type: none"> • A (ABA number) • C (CHIPS UID) • S (S.W.I.F.T. ID)
KEY_ACC (pk)	(varchar 30). Account number.

AUX_TIME_STAMP (pk)	(varchar 16). Date and time when the auxiliary address record was updated.
BANK (fk)	(varchar 3). Bank identifier of the bank that owns this address.
KEY_ADR	(varchar 5). Address code used to distinguish among multiple addresses sharing the same account.
KEY_PAD	(varchar 1). Number of additional lines for addresses when a bank has several addresses.
DAT_FUNCTION	(varchar 7). Function that last accessed this account record; contains following values: <ul style="list-style-type: none"> • VFY ADD (Verify add) • VFY DEL (Verify delete) • VFY UPD (Verify update)
CREATOR	(varchar 10). Operator ID and bank ID of the operator who created the address record.
CREATE_DATE	(date). Date that the address record was created.
UPDATER	(varchar 10). Operator ID and bank ID of the operator who updated the address record.
UPDATE_DATE	(date). Date when the address record was updated.
VERIFIER	(varchar 10). Operator ID and bank ID of the operator who verified the address record.
VERIFY_DATE	(date). Date when the address record was verified.
DAT_OWNER_PROD_ID	(varchar 3). Product ID.
DAT_OWNER_BANK_ID (fk).	(varchar 3). Bank ID.
DAT_OWNER_LOC	(varchar 6). Location ID.
DAT_OWNER_CUST	(varchar 20). Customer ID.
LAST_USED	(date). Date on which this record was used last.
SPECIAL_IDBANK	(varchar 3). Bank ID.
SPECIAL_IDKEY	(varchar 20). Bank name.
CONCEN_BANK_ID (fk).	(varchar 3). Concentration account bank ID.
CONCEN_ID_TYPE (fk).	(varchar 1). Concentration account type.

Auxiliary Tables

CONCEN_ACCOUNT (fk).	(varchar 30). Concentration account number.
GRP_ID (fk).	(varchar 10). Group ID.
WE_INHIB_THEM	(varchar 1). CHIPS bilateral information.
THEY_INHIB_US	(varchar 1). CHIPS bilateral information.
ERROR_INHIB	(varchar 1). Indicates whether an error exists with CHIPS bilateral information.
CHP_RCV_LIMIT	(number 21.3). Incoming CHIPS message limit. Messages exceeding this amount go to risk.
CHP_SND_LIMIT	(number 21.3). Outgoing CHIPS message limit. Messages exceeding this amount go to risk.
ACC_MINIMUM	(number 21.3). Minimum start-of-day account balance.
ACC_SOD_CHF	(number 21.3). Start-of-day clearinghouse funds balance.
ACC_SOD_LEDGER	(number 21.3). Start-of-day ledger balance.
ACC_SOD	(number 21.3). Account start-of-day balance.
GRP_MINIMUM	(number 21.3). Minimum start-of-day balance for a group.
GRP_SOD_CHF	(number 21.3). Start-of-day clearinghouse funds balance for a group.
GRP_SOD_LEDGER	(number 21.3). Start-of-day ledger balance for a group.
GRP_SOD	(number 21.3). Start-of-day balance for a group.
SNAME_ID	(varchar 30). Short name of bank; can be in the form shortname/citycode.
ABA_ID	(varchar 9). American Bankers Association routing identification number.
SWIFT_ID	(varchar 30). 8- or 11-character S.W.I.F.T. ID.
BRANCH_ID	(varchar 30). Not populated in this table.
CHIPS_UID_ID	(varchar 30). CHIPS user identification number.
USER_ID	(varchar 30). Not populated in this table.
INTERBNK_ID	(varchar 30). Not populated in this table.
CUSTOMER_ID	(varchar 30). Not populated in this table.
CUSTOMER_NAME	(varchar 35). Not populated in this table.

ADR_NAME	(varchar 35). Address name.
ADR1	(varchar 35). First line of the mailing address.
ADR2	(varchar 35). Second line of the mailing address.
ADR3	(varchar 35). Third line of the mailing address.
ZIP	(varchar 10). Not populated in this table.
COUNTRY_CODE	(varchar 2). Not populated in this table.
PHN	(varchar 20). Not populated in this table.
EXT	(varchar 4). Not populated in this table.
SPEED	(varchar 4). Not populated in this table.
ADR_DESC	(varchar 20). Not populated in this table.
ADR_CLASS	(varchar 6). Not populated in this table.
ADR_CITY	(varchar 15). Not populated in this table.
ADR_DBT_DEPT	(varchar 20). Not populated in this table.
ADR_CDT_DEPT	(varchar 20). Not populated in this table.
ADR_LOCATION	(varchar 6). Not populated in this table.
DBT_RT_CODE	(varchar 6). Not populated in this table.
CDT_RT_CODE	(varchar 6). Not populated in this table.
CUSTOMER_CODE	(varchar 10). Not populated in this table.
FED_SHORT_NAME	(varchar 20). Federal Reserve short name; for an ABA account only.
FED_ONLINE_FLAG	<p>(varchar 1). Federal Reserve online status flag for ABAs only; contains the following values:</p> <ul style="list-style-type: none"> • Y (Online) • N (Not online) • F (Foreign) • I (Not eligible) • S (Settlement)
DBT_SPEC_INSTR1	(varchar 35). Not populated in this table.

Auxiliary Tables

DBT_SPEC_INSTR2	(varchar 35). Not populated in this table.
DBT_SPEC_INSTR3	(varchar 35). Not populated in this table.
CDT_SPEC_INSTR1	(varchar 35). Not populated in this table.
CDT_SPEC_INSTR2	(varchar 35). Not populated in this table.
CDT_SPEC_INSTR3	(varchar 35). Not populated in this table.
PHONE_TEST_TYPE	(varchar 1). Not populated in this table.
PHONE_EARLY_HRS	(varchar 2). Not populated in this table.
PHONE_EARLY_MNS	(varchar 2). Not populated in this table.
DOCUMENT_ID	(varchar 6). Not populated in this table.
RISK_COUNTRY	(varchar 2). Not populated in this table.
DEBIT_AUTH_FLAG	(varchar 1). Not populated in this table.
LOCAL_FLAG	(varchar 1). Not populated in this table.
PRIORITY_FLAG	(varchar 1). Not populated in this table.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
TST_PARENT_REL_ID	(number 10). Not populated in this table.
TST_CONTACT	(varchar 20). Not populated in this table.
TEST_PHONE_EXT	(varchar 4). Not populated in this table.
TEST_PHONE_NUM	(varchar 12). Not populated in this table.
TST_INSTR1	(varchar 35). Not populated in this table.
TST_INSTR2	(varchar 35). Not populated in this table.
DORMANT_FLAG	(varchar 1). Not populated in this table.
SLOTS_SAVED	(number 5). Not populated in this table.
SERIAL_RESET_FLAG	(varchar 1). Not populated in this table.
GAP_BO_LIMIT	(number 5). Not populated in this table.

GAP_BS_LIMIT	(number 5). Not populated in this table.
GAP_BA_LIMIT	(number 5). Not populated in this table.
GAP_FO_LIMIT	(number 5). Not populated in this table.
GAP_FS_LIMIT	(number 5). Not populated in this table.
GAP_FA_LIMIT	(number 5). Not populated in this table.
IN_GENERATE_FLAG	(varchar 1). Not populated in this table.
IN_ACK_FIX_DATE	(date). Not populated in this table.
IN_FIX_PERSON	(varchar 10). Not populated in this table.
IN_PREV_ALG_ID	(varchar 12). Not populated in this table.
IN_PREV_FIX_DATE	(date). Not populated in this table.
IN_CUR_ALG_ID	(varchar 12). Not populated in this table.
IN_CUR_FIX_DATE	(date). Not populated in this table.
IN_NEXT_ALG_ID (fk).	(varchar 12). Not populated in this table.
IN_NEXT_FIX_DATE	(date). Not populated in this table.
OUT_GENERATE_FLAG	(varchar 1). Not populated in this table.
OUT_ACK_FIX_DATE	(date). Not populated in this table.
OUT_FIX_PERSON	(varchar 10). Not populated in this table.
OUT_PREV_ALG_ID	(varchar 12). Not populated in this table.
OUT_PREV_FIX_DATE	(date). Not populated in this table.
OUT_CUR_ALG_ID	(varchar 12). Not populated in this table.
OUT_CUR_FIX_DATE	(date). Not populated in this table.
OUT_NEXT_ALG_ID	(varchar 12). Not populated in this table.
OUT_NEXT_FIX_DATE	(date). Not populated in this table.
SEC_ONLINE_FLAG	<p>(varchar 1). Federal Reserve online status flag for securities only; contains the following values:</p> <ul style="list-style-type: none"> Y (Online)

Auxiliary Tables

	<ul style="list-style-type: none">• N (Not online)• I (Not eligible)
PENDING_DEL_DATE	(date). Date when the auxiliary address record was deleted. A deleted address record remains in the database until the period is purged.
CHIPS_OPEN_HH	(varchar 2). Hour portion of time that CHIPS processing opens.
CHIPS_OPEN_MM	(varchar 2). Minute portion of time that CHIPS processing opens.
FED_OPEN_HH	(varchar 2). Hour portion of time that FED processing opens.
FED_OPEN_MM	(varchar 2). Minute portion of time that FED processing opens.
FX_TRADE_REL_TYPE	(varchar 1). Foreign exchange relationship type.
TIME_ZONE_NAME	(varchar 10). Name of the time zone.
SRV_MSG_FLAG	(varchar 1). Indicates whether to send a service message (notification). to the third party.
INC_HDR_FLAG	(varchar 1). Indicates whether to include a line of text in the message's header describing the third party test.
SWF_SUBTYPE	(varchar 4). SWIFT participant subtype.
PROFILE_IDBANK	(varchar 3). Differentiator to allow configurable cross bank indices.
PROFILE_IDTYPE	(char 1). ID type for the profile.
PROFILE_IDACC	(varchar 30). Account number for the account type address indices.
PROFILE_IDADR	(varchar 5). Address code for the account type address indices.
PROFILE_IDPAD	(char 1). Padding for the account type address indices.

Auxiliary ABA DDA Table (AUX_ABA_DDA_T)

The Auxiliary ABA/DDA table contains CHIPS participant information for CHIPS UIDs as defined on the AUX maintenance screen.

Primary Keys

The Auxiliary ABA DDA table (AUX_ABA_DDA_T) has the following primary keys:

- TYPE
- KEY_ACC
- PARTICIIPANT_ABA
- AUX_TIME_STAMP

Column Descriptions

TYPE (pk)	(varchar 1). Address ID type. This column is part of a local, bank-defined customer information file. This column contains the following values: A (ABA number) C (CHIPS UID) S (S.W.I.F.T. ID)
KEY_ACC(pk)	(varchar 30). Account number.
PARTICIPANT_ABA(pk)	(varchar 4). Participant's ABA identification.
AUX_TIME_STAMP (pk)	(varchar 16). Date and time when the auxiliary address record was updated.
PARTICIPANT_DDA	(varchar 34). Participant's DDA number.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Auxiliary Notes Table (AUX_NOTES_T)

The Auxiliary Notes table contains the notes, or descriptions, of the AUX entered on the AUX maintenance screen.

Primary Keys

The Auxiliary Notes table (AUX_NOTES_T) has the following primary keys:

- TYPE
- KEY_ACC
- SERIAL_ID
- AUX_TIME_STAMP

Column Descriptions

TYPE (pk)	(varchar 1) Address ID type. This column is part of a local, bank-defined customer information file. This column contains the following values: <ul style="list-style-type: none">• A (ABA number)• C (CHIPS UID)• S (S.W.I.F.T. ID)
KEY_ACC(pk)	(varchar 30) Account number.
SERIAL_ID(pk)	(number 19) A numeric ID that uniquely identifies this note.
AUX_TIME_STAMP (pk)	(varchar 16) Date and time when the auxiliary address record was updated.
NOTE_LINE	(number 5) Sequence number of the line number of the note.
NOTE	(varchar 80) Text that describes the AUX.
RECORD_EXPIRED	(varchar 16) Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1) Indicates whether this row's timestamp column has been propagated to previous rows.

Anticipation Template Tables

The Anticipation Template tables contain debit, credit and destination information for anticipation templates.

MTS provides the ability for operators to create, maintain, display, delete, and verify anticipation templates. Operators enter default data for anticipations onto the templates. When the details of the expected funds are available, the operator creates one or more anticipations from an Anticipation Template by entering the Amount and the Originator's Reference.

Parent/child relationships

The Anticipation Template table (ANT_TMPL_T) is the parent table to the Anticipation Template Credit () table and the Anticipation Template Debit (ANT_TMPL_DR_T) table.

Primary keys

The Anticipation Template tables have these primary keys in common:

- ANT_TMPL_ID
- ANT_TIME_STAMP

Table Relationships

All Anticipation Template tables relate to each other through the ANT_TMPL_ID column. Anticipation tables are stand-alone and do not relate to any other tables in the data dictionary.

Anticipation Template Table (ANT_TMPL_T)

The Anticipation Template table is a holding place for data from anticipation messages.

This table works in conjunction with the and ANT_TMPL_DR_T tables for each anticipation.

Table Relationships

The Anticipation Template table [ANT_TMPL_T] is the parent table to the Anticipation Template Credit table and the Anticipation Template Debit [ANT_TMPL_DR_T] table. This table relates to the other anticipation template tables, that is, and ANT_TMPL_DR_T tables, through the ANT_TMPL_ID column.

Column Descriptions

ANT_TMPL_ID (pk)	varchar (20) Anticipation template ID.
ANT_TIME_STAMP (pk)	varchar (16) Record update time.
DAT_FUNCTION	varchar (7) Database maintenance event that triggered the data export.
CREATOR	varchar (10) Operator who created the anticipation template.
CREATE_DATE	date. Date on which the anticipation template was created.
UPDATER	varchar (10) Operator who last updated the anticipation template.
UPDATE_DATE	date. Date on which the anticipation template was last updated.
VERIFIER	varchar (10) Operator who verified the anticipation template. The template may have been added or updated.
VERIFY_DATE	date. Date on which the anticipation template was verified.
DAT_OWNER_PROD_ID	varchar (3) Product ID.
DAT_OWNER_BANK_ID	varchar (3) Bank ID.
DAT_OWNER_LOC	varchar (6) Location ID.
DAT_OWNER_CUST	varchar (20) Customer ID.
MSGTYPE	varchar (9) Type of message union.
DESCRIPTION	varchar (80) Description.

EXPIRATION_DATE	date. Date on which the anticipation template either expires or is no longer in effect. A value of 0 (zero) indicates the template never expires.
EXPIRATION_TIME	varchar (5). Time when the anticipation template either expires or is no longer in effect. A value of 0 (zero) indicates the template never expires.
RECORD_EXPIRED	varchar (16). Date and time when this record becomes invalid.
RECORD_UPDATED	varchar (1). Indicates whether this row has had its timestamp propagated.
TRN_DATE	date. Processing date portion of the transaction number indicating the date when the message was created; from the TRN (Transaction Reference Number).
TRN_NUMBER	number (8). Serial number portion of the transaction number. Unique serial number assigned to the message by REFSERV (a background process in MTS that assigns the second half of the next consecutive transaction number). The first transaction message of each processing day is 00000001.
OWN_BANK	varchar (3). Owner bank name.
OWN_LOC	varchar (6). Owner location.
PROXY	varchar (6). Proxy location.
BANK	varchar (3). Bank ID.
TYPE	varchar (1). ID type for the account indices.
KEY_ACC	varchar (30). Account number for the account type address indices.
KEY_ADR	varchar (5). Address code for the account type address indices.
KEY_PAD	varchar (1). Padding for the account type address indices.
REPETITIVE_ID	varchar (8). Repetitive ID. Unique identifier assigned to the repetitive template that MTS uses to generate an outgoing transaction.
SOURCE_CD	<p>varchar (3). Transaction Source Code; mapped from the SRC field on the Funds Transfer (FTR) screen. All sources are defined in configuration files. This column contains the following values:</p> <ul style="list-style-type: none"> • ADJ (Adjustment) • ADM (Administrative message entry function) • CHP (CHIPS) • CMS (Cash management system) • DFM (Due from monitor) • DOC (Letter of credit) • ENT (Payment entry function) • ETW (Enhanced Trading Workstation)

Anticipation Template Tables

	<ul style="list-style-type: none"> • FED (Fedwire) • INT (Internal transfer) • LTC (Letter requiring callback) • LTR (Letter) • MEM (Memo past function) • MIS (Miscellaneous, for example: cash management system or remote batch entry) • MTW (Money Transfer Workstation) OFL (Offline) • PHN (Phone transfer initiated outside bank) • RBE (Remote batch entry) • SOD (Start-of-day balance load) • STO (Standing order function) • SWF (S.W.I.F.T.) • WIR (Telex) <p>This column also applies to any user-defined source.</p>
TYPE_CD	varchar (4). Type code mapped from the MTP field on the Funds Transfer (FTR) screen. Outgoing Federal Reserve, S.W.I.F.T., or CHIPS message type; correlates to the MSG_TYPE_TYP column in the Message Extract File (MFE).
SUBTYPE	varchar (4). Subtype code.
VALUE_DATE	date. Current, future, or back-dated transaction date; mapped from the VAL field on the FTR screen.
TRAN_TYPE	varchar (3). The transaction type: FTR, FFR, FFS, DEP, DRW, DFT, DFA, ANT, NON, PRE, SPL.
FUNDS_TYPE	varchar (1). Type of funds; mapped from the FNDS field on the FTR screen. This column contains the following values: N = next day S = same day
INCOMING_MSGTYPE	varchar (4). The incoming message type.
INCOMING_FORMAT	varchar (4). The incoming message format.
INCOMING_REF	varchar (80). The external input reference: FED = IMAD/OMAD CHIPS = ISN/SSN/OSN SWIFT = MIR/MOR
FRONTEND_REF_NUM	varchar (16). The bank's internal system reference number.
DOC_NUM	varchar (7). Batch document number.

ITEM_NUM	varchar (2). Batch item number.
FUNDS_CATEGORY	varchar (10). Batch fund's category.
INSTR_ADV_TYPE	varchar (3). Advice method indicated by the sender.
RECEIVE_CHAR	number (10). Number of characters in the received message.
SEGMENT_INDEX	number (10). The value is non-zero if this message was split from a segmented Telex.
SPLIT_CTR	number (5). Number of MT102 child transactions created.
RPR_LEVEL	varchar (2). The level of parsing performed in the incoming mapper.
RECEIPT_TIME	date. Instruction received time, as indicated by the originating system.
INST_DATE	date. Instruction received date, as indicated by MTS.
INST_TIME	date. Instruction received time, as indicated by MTS.
POST_DATE	date. Debit posting date.
SEND_DATE	date. Date on which the advice is to be sent.
RELEASE_TIME_HH	varchar (2). Release time hours
RELEASE_TIME_MM	varchar (2). Release time minutes.
POST_TIME	date. Debit posting time.
AMOUNT	number (21.3). Amount for which the anticipation was created.
BASE_AMOUNT	number (21.3). Base currency amount.
CURRENCY_CODE	varchar (3). Currency code.
EXCH_RATE	number (29.11). Not applicable to Anticipations.
FEDFUND_TYPE	varchar (9). Fed funds type.
FED_DAYS	number (5). Number of days for a fed funds message.
INTEREST_RATE	number (18.5). Interest rate for a fed funds message.
TOTAL_AMOUNT	number (21.3). Transaction accounting.
ANT_TOL_AMT	number (21.3). Tolerance amount for anticipation matching.
CUSIP	varchar (9). CUSIP ID for a particular security's TRN.

Anticipation Template Tables

CUSIP_DESC	varchar (20). CUSIP description for a particular security's TRN.
CAL	varchar (4). Caller's internal telephone extension.
CAL_OVR	varchar (1). Caller override flag. Set to * during entry when the sending bank's branch and the debit party are left blank. * indicates there was no caller.
CALLER	varchar (35). Name of the caller who authorized the transfer.
PIN_REF_ID	number (10). PIN reference ID.
VFC_PIN_REF_ID	number (10). PIN of the previous verify callback.
PHN_CUS_BANK	varchar (3). Bank identifier of the bank that owns this anticipation record.
PHN_CUS_TYPE	varchar (1). ID type for the account indices.
PHN_CUS_KEY_ACC	varchar (30). Account number for the account type address indices.
PHN_CUS_KEY_ADR	varchar (5). Address code for the account type address indices.
PHN_CUS_KEY_PAD	varchar (1). Padding for the account type address indices.
TRADER_CTRL	varchar (10). Trader control number.
TESTKEY_IN	varchar (16). Testkey in.
TEST_TEXT	varchar (24). Used to compute the test key.
RISKINESS_IND	varchar (2). Riskiness indicator.
DBT_CHRG	varchar (1). Flag indicating whether charges apply for debit; mapped from the CHG:DB field on the FTR screen. Values are bank-defined, determined from user-defined configuration routines for fee charges.
CDT_CHRG	varchar (1). Flag indicating whether charges apply for credit; mapped from the CHG:CD field on the FTR screen. Values are bank-defined, determined from user-defined configuration routines for fee charges.
COMMISSION	varchar (1). Flag indicating whether charges apply for commission; mapped from the CHG:COM field on the Funds Transfer (FTR) screen. Values are bank-defined, determined from user-defined configuration routines for commission charges.
CBL_CHARGE	varchar (1). Flag indicating whether charges are to be applied for cable; mapped from the CHG:CAB field on the Funds Transfer (FTR) screen. Values are bank-defined, determined from user-defined configuration routines for cable charges.
COMM_MODE	varchar (1). Commission charge mode.
CBL_MODE	varchar (1). Cable charge mode.

DEST_COUNT	number (5). Destination count.
DEST_FUNC	varchar (3). Destination function: DOC, FUT, HLD, PAY, RPR, RSK, SUP, VFY
DEST_Q_CLASS	varchar (3). Identifies the Q class: BLT, BKW, ENT, FED, INT, SWF, WIR
DEST_OPR	varchar (3). Operator initials for the initialed queue.
ADM_LOCK_TEXT	varchar (1). Set when in fill-in-the-blanks mode.
BNK_TX	varchar (1). Bank TX flag.
MULT_DBT	varchar (1). Set when there are multiple debit parties.
MULT_CDT	varchar (1). Set when there are multiple credit parties.
AMT_VFY	varchar (1). Setting requires key verifying the amount.
CUR_VFY	varchar (1). Setting requires key verifying the currency.
EXCH_VFY	varchar (1). Setting requires key verifying the exchange rate.
SERIAL_VFY	varchar (1). Set when key verify is required.
CALL_NON_RPT	varchar (1). Set by data entry when the operator overrides caller data.
CALL_NOF	varchar (1). Set by data entry when the operator overrides caller data.
CALL_LIM	varchar (1). Set by data entry when the operator overrides caller data.
CALL_RPT	varchar (1). Set by data entry when the operator overrides caller data.
CALLBACK	varchar (1). Set by data entry when the operator overrides caller data.
FAIL_TST	varchar (1). Set by data entry when the operator overrides a failed test.
RPTV_LIM	varchar (1). Set by data entry when the operator overrides repetitive data.
RPTV_CHNG	varchar (1). Set by data entry when the operator overrides repetitive data.
PDM	varchar (1). Set when any part of the message is being redelivered on an output line.
ADV_OVR	varchar (1). Set by the input mapper if overriding the default advising.
TEST_REQ	varchar (1). Test type, if required: N = No test

Anticipation Template Tables

	A = S.W.I.F.T. authentication S = Serial test Y = Test required
ACCT_VFC	varchar (1). Account callback limit exception; Rep or Non-rep.
CDT_ADV	varchar (1). Set by data entry when the operator overrides default data.
MULTI_CUR	varchar (1). Multi-currency flag.
PRIORITY	varchar (1). Current priority flag.
DELIVERY	varchar (1). Delivery notification flag.
MON_INTERCEPT	varchar (1). Flag set by RISK_AND_ROUTING if an intercept monitor point is matched.
NOTIFY	varchar (1). Originator notification flag.
STOP_INTERCEPT	varchar (1). Set by STOP_CHECK if an intercept stop descriptor is matched.
RTP_INTERCEPT	varchar (1). Set by message routing if a release time payment has been held (in Risk).
MATCHED_ANT	varchar (1). Matched anticipation flag.
REPAIR_CHNG	varchar (1). Set at Repair if any credit or debit data is changed.
CHIPS_RECVRY	varchar (1). Chips recovery in progress.
NON_ACCTING	varchar (1). Value is Y if the message is a non-accounting memo post.
SWF_STMT	varchar (1). Value is Y if the message should be included in S.W.I.F.T. statements.
CHARGE_STATE	varchar (1). Value is T if the message is charged as straight-through.
VFY_COUNT	varchar (1). Number of times the message has passed through function VFY (Verify). , not counting function HVY (High Amount Verify).
MPA_DBT_AUTH	varchar (1). Value is Y if the debit is authorized by the external DDA system.
MPA_DBT_POST	varchar (1). Value is Y if the debit is posted to the external DDA system.
MPA_CDT_AUTH	varchar (1). Value is Y if the credit is authorized by the external DDA system.
MPA_CDT_POST	varchar (1). Value is Y if the credit is posted to the external DDA system.
TEST_STATE	varchar (1). Test or authorization result. P = Test placed or good authenticator O = Test or authentication succeeded with old key

	<p>N = Test or authentication succeeded with next key</p> <p>F = Test or authentication failed</p> <p>B = Test or authentication was bypassed</p> <p>U = Test or authentication relationship unknown</p> <p>Y = Input mapper did not send message to repair/verify</p>
STRAIGHT_THRU	varchar (1). Flag indicating whether the message was processed straight through. The flag may be set to any value the bank has in its route.cfg fully parse column.
PAY_STATE	<p>varchar (1). Accounting pay state flag indicating which type of accounting has been performed. This column contains the following values:</p> <p>Blank (No accounting performed).</p> <p>P (Normal accounting performed).</p> <p>D (Debit side of split accounting performed).</p> <p>C (Credit side of split accounting performed).</p> <p>H (Debit normal, credit to holdover account).</p> <p>R (Debit to holdover account, credit normal).</p>
EXCH_RATE_FLG	<p>varchar (1). Exchange rate flag:</p> <p>I = Indirect V = Assigned on value date Blank = Within tolerance</p>
FX_TOLERANCE	<p>varchar (1). Foreign e xchange tolerance:</p> <p>Y = Exceeds tolerance O = Overridden</p>
VFC_COUNT	varchar (1). Number of times the message has passed through function VFC (Verify Callback). A value of Blank indicates the callback has not yet been made.
FUNDED_MSG	varchar (1). Indicates whether the message is MPA (Mainframe Posted and Authorization) funded.
PMT_CHRG	varchar (1). Payment charge flag.
INT_REROUTE	varchar (1). Indicates whether the message is internally re-routed.
CUR_HOLD	varchar (1). Value is Y if the hold flag for the currency is set. Otherwise, the value is blank.
OVR_OUT_LIM	varchar (1). Value is Y if an additional currency rate review is required.
FORCE_POST	varchar (1). Value is Y if the message has been force-posted.
RPTD_SWF_STMT	varchar (1). Value is Y if the transaction is already reported on the S.W.I.F.T. statement.
DBT_CNF_OVER	varchar (1). Flag indicating whether the operator has overridden the debit confirmation (*) or not (blank).
DBT_ADV_OVER	varchar (1). Flag indicating whether the operator has overridden the debit advice (*) or not (blank).

Anticipation Template Tables

CDT_CNF_OVER	varchar (1). Flag indicating whether the operator has overridden the credit confirmation (*) or not (blank).
CDT_ADV_OVER	varchar (1). Flag indicating whether the operator has overridden the credit advice (*) or not (blank).
ALT_TO_RSK	varchar (1). Flag indicating whether the payment is to be sent to risk if the RPT template is altered during payment entry (*) or not (blank).
VFY_REQ	varchar (1). Flag indicating whether the payment is to be routed as a non repetitive (*) or not (blank).
ALLOW_CHANGES	varchar (1). Value is Y if the entry operator is allowed to make changes.
INTERNAL	varchar (1). Value is Y if the repetitive was requested internally.
NOMODIFY	varchar (1). Value is Y if the BBI and the OBI may not be changed.
PRE_FOUND	varchar (1). Value is Y if the PRE standing instruction is found.
FX_FOUND	varchar (1). Value is Y if the foreign exchange standing instruction is found.
CHECK_FOUND	varchar (1). Value is Y if the Check standing instruction is found.
COR_DONE	varchar (1). Value is Y if the COR standing instruction was executed and accepted.
RTE_DONE	varchar (1). Indicates that routing is completed.
PMT_CHAR_CODE	varchar (10). The payment characteristics code. Code that indicates the payment characteristics to the channel. In Belgium, this is called a Nature Code.
FIN_CPY_SRC_FLG	varchar (1). Value R indicates message type 102 is a request for credit advice.
REG_ID_TYPE	varchar (1). Regulatory ID type.
REG_ID	varchar (34). Regulatory ID.
BOPR_TEXT1	varchar (35). Balance of payment reporting justification text line1.
BOPR_TEXT2	varchar (35). Balance of payment reporting justification text line 2.
BOPR_TEXT3	varchar (35). Balance of payment reporting justification text line 3.
MATCH_DBT_ID	char (1). Anticipation match debit ID.
MATCH_CDT_ID	char(1). Anticipation match credit ID.
MATCH_BNP_ID	char (1). Anticipation match beneficiary ID.
MATCH_SRN	char (1). Anticipation match sender reference number.

MATCH_SIDE	char (1). Automated partial matching, for the anticipation.
UMBRELLA_FLG	char (1). Indicates which side of the anticipation to match.

Anticipation Template Credit Table (ANT_TMPL_CR_T)

The Anticipation Template Credit table is a holding place for data from the credit side of anticipation messages.

This table works in conjunction with the ANT_TMPL_T and ANT_TMPL_DR_T tables for each anticipation.

Primary Keys

The Anticipation Template Credit table (ANT_TMPL_CR_T). has the following primary keys:

- ANT_TMPL_ID
- ANT_TIME_STAMP

Table Relationships

This table relates to the other anticipation template tables, that is, ANT_TMPL_T and ANT_TMPL_DR_T tables, through the ANT_TMPL_ID column.

Column Descriptions

ANT_TMPL_ID (pk).	varchar (20). Anticipation template ID.
ANT_TIME_STAMP (pk).	varchar (16). Record update time.
RECORD_EXPIRED	varchar (16). Date and time when this record becomes invalid.
RECORD_UPDATED	varchar (1). Indicates whether this row has had its timestamp propagated.
CDT_VALUE_DATE	dateCredit value date.
CDT_ADR_BNK_ID	varchar (3). Credit party address bank ID.
CDT_REL_ID	number (10). Credit party address reference ID.
CDT_OVR	varchar (1). Credit party override.
CDT_IDTYPE	varchar (1). Credit party ID type.
CDT_ID	varchar (64). Credit party ID.

BANK_ID	varchar (3). Differentiator to allow configurable cross bank indices.
ID_TYPE	varchar (1). Account ID type; contains the following values: <ul style="list-style-type: none"> • D (DDA) • F (Foreign nostro account number) • G (General ledger account number) • P (CHIPS participant ID) • V (Savings account number)
ACCOUNT	varchar (30). Account number.
CDT_AMOUNT	number (21.3). Credit dollar value of the transaction; mapped from the AMT field on the Funds Transfer (FTR). screen.
CDT_CURRENCY	varchar (3). When a multi-currency message.
SEC_FLG	varchar (16). Secondary account flag.
SEC_AMT	number (21.3). Secondary amount, either charges or commissions.
SEC_CUR	varchar (3). Secondary currency.
SEC_ACC_BANK_ID	varchar (3). Differentiator to allow configurable cross bank indices.
SEC_ACC_ID_TYPE	varchar (1). Secondary account ID type; contains the following values: <ul style="list-style-type: none"> • D (DDA) • F (Foreign nostro account number) • G (General ledger account number) • P (CHIPS participant ID) • V (Savings account number)
SEC_ACC_ACCOUNT	varchar (30). Account number that identifies the secondary account.
TER_FLG	varchar (16). Tertiary account flag.
TER_TER_AMT	number (21.3). Tertiary amount, either charges or commissions.
TER_TER_CUR	varchar (3). Tertiary currency.
TER_ACC_BANK_ID	varchar (3). Differentiator to allow configurable cross bank indices.
TER_ACC_ID_TYPE	varchar (1). Tertiary account ID type; contains the following values: <ul style="list-style-type: none"> • D (DDA)

Anticipation Template Tables

	<ul style="list-style-type: none"> • F (Foreign nostro account number) • G (General ledger account number) • P (CHIPS participant ID) • V (Savings account number)
TER_ACC_ACCOUNT	varchar (30). Account number that identifies the tertiary account.
CDT_ACC_CITY	varchar (15). Credit account city.
CDT_ACC_CLASS	varchar (6). Credit account class.
CDT_ACC_COUNTRY	varchar (2). Credit account country.
CDT_ACC_PARENT	varchar (10). Credit account parent code.
CDT_ACC_PRODS	varchar (20). Credit account product codes.
CDT_SYS_OF_REC	varchar (3). Credit system of record flag. If credit authorization is from another system, this column has a value.
CONCEN_BANK_ID	varchar (3). Differentiator to allow configurable cross bank indices.
CONCEN_ID_TYPE	varchar (1). Concentration account ID type; contains the following values: <ul style="list-style-type: none"> • D (DDA) • F (Foreign nostro account number) • G (General ledger account number) • P (CHIPS participant ID) • V (Savings account number)
CONCEN_ACCOUNT	varchar (30). Concentration account number.
CDT_PARENT_CODE	varchar (10). Credit account parent code. A parent code is a method of grouping accounts for special processing (such as end-of-day processing in MFE). MTS does not process this column.
CDT_ADR_CLASS	varchar (6). Credit address class.
CDT_ADR_CITY	varchar (15). Credit address city.
CDT_ADR_COUNTRY	varchar (2). Credit address country.
CDT_ADR_TYPE	varchar (6). Credit address type.
CDT_ADV_TYP	varchar (3). Credit advice type.

CDT_ADR_ADV_TYPE	varchar (3). Advice method indicated by credit address, and possibly overridden by the Repair operator.
CDT_DEPARTMENT	varchar (20). Credit department.
CDT_LOCATION	varchar (6). Credit location.
CDT_RT_CODE	varchar (6). Credit routing code.
CDT_RT_STATE	varchar (1). Indicates the routing code instructions are queued. Q = Routing code instructions are queued
CDT_SEC_SUB_ACCT	varchar (61). Credit or sender's sub-account (in the case of a securities transaction).
CDT_SHNAM	varchar (20). Fed short name. If blank, then the first line of the name is used by the mapper.
IDTYPE	varchar (1). ID type for party (address) indices.
IDKEY	varchar (64). Name, or routing ID, or account number/address code.
ADV_TYPE	varchar (3). Final payment advice type.
PHN_TIME	varchar (4). Final payment phone time.
PYMNT_Q_PROD_ID	varchar (3). Product ID.
PYMNT_Q_BANK_ID	varchar (3). Bank ID.
PYMNT_Q_LOC	varchar (6). Location ID.
PYMNT_Q_CUST	varchar (20). Customer ID.
PYMNT_Q_NAME	varchar (33). Queue name.
CDT_NAME1	varchar (35). Credit name line 1.
CDT_NAME2	varchar (35). Credit name line 2.
CDT_NAME3	varchar (35). Credit name line 3.
CDT_NAME4	varchar (35). Credit name line 4.
CDT_ADV_INST1	varchar (35). Credit advice instruction line 1.
CDT_ADV_INST2	varchar (35). Credit advice instruction line 2.
CDT_ADV_INST3	varchar (35). Credit advice instruction line 3.

Anticipation Template Tables

CDT_RECON_REF	varchar (12). Credit reconciliation reference.
CDT_SPC_INST1	varchar (35). Credit special instruction line 1.
CDT_SPC_INST2	varchar (35). Credit special instruction line 2.
CDT_SPC_INST3	varchar (35). Credit special instruction line 3.
CDT_HOLD_FLG	varchar (1). Credit party hold flag.
CDT_PAY_FLG	varchar (1). State of payment instruction.
CDT_SECWIR_FLG	varchar (1). State of secondary wire instruction.
CDT_ALT_SWIFT_FLG	varchar (1). Specify a payment routing S.W.I.F.T. ID from an advice type standing instruction.
CDT_NO_PHNADV_FLG	varchar (1). Indicates whether the NO NEED TO PHONE command was invoked in the PHN function.
CDT_DELIVERY_FLG	varchar (1). Credit delivery flag.
CDT_NOF_LOOKED_UP	varchar (1). Indicates whether the not-on-file credit party has been looked up. T = not-on-file credit party was looked up
CDT_ID_CHG_FLG	varchar (1). Indicates whether the credit ID has changed. Y = credit ID was changed
IB1_ADR_BNK_ID	varchar (3). First intermediary bank party address bank ID.
IB1_REL_ID	number (10). First intermediary bank party address reference ID.
IB1_SECWIR	varchar (1). First intermediary bank party secondary wire.
IB1_IDTYPE	varchar (1). First intermediary bank party ID type.
IB1_ID	varchar (64). First intermediary bank party ID.
IB1_NAME1	varchar (35). First intermediary bank party line 1.
IB1_NAME2	varchar (35). First intermediary bank party line 2.
IB1_NAME3	varchar (35). First intermediary bank party line 3.
IB1_NAME4	varchar (35). First intermediary bank party line 4.
IB1_ADV_INST1	varchar (35). First intermediary bank party advice line 1.
IB1_ADV_INST2	varchar (35). First intermediary bank party advice line 2.

IB1_ADV_INST3	varchar (35). First intermediary bank party advice line 3.
IBK_ADR_BNK_ID	varchar (3). Intermediary bank party address bank ID.
IBK_REL_ID	number (10). Intermediary bank party address reference ID.
IBK_SECWIR	varchar (1). Intermediary bank party secondary wire.
IBK_IDTYPE	varchar (1). Intermediary bank party ID type.
IBK_ID	varchar (64). Intermediary bank party ID.
IBK_NAME1	varchar (35). Intermediary bank party name line 1.
IBK_NAME2	varchar (35). Intermediary bank party name line 2.
IBK_NAME3	varchar (35). Intermediary bank party name line 3.
IBK_NAME4	varchar (35). Intermediary bank party name line 4.
IBK_ADV_INST1	varchar (35). Intermediary bank party advice instruction line 1.
IBK_ADV_INST2	varchar (35). Intermediary bank party advice instruction line 2.
IBK_ADV_INST3	varchar (35). Intermediary bank party advice instruction line 3.
BBK_ADR_BNK_ID	varchar (3). Beneficiary bank party address bank ID.
BBK_REL_ID	number (10). Beneficiary bank party address reference ID.
BBKF_SECWIR	varchar (1). Beneficiary bank party secondary wire.
BBKF_IDTYPE	varchar (1). Beneficiary bank party ID type.
BBKF_ID	varchar (64). Beneficiary bank party ID.
BBK_NAME1	varchar (35). Beneficiary bank party name line 1.
BBK_NAME2	varchar (35). Beneficiary bank party name line 2.
BBK_NAME3	varchar (35). Beneficiary bank party name line 3.
BBK_NAME4	varchar (35). Beneficiary bank party name line 4.
BBK_ADV_INST1	varchar (35). Beneficiary bank party advice line 1.
BBK_ADV_INST2	varchar (35). Beneficiary bank party advice line 2.
BBK_ADV_INST3	varchar (35). Beneficiary bank party advice line 3.

Anticipation Template Tables

BNP_CHARGE_FLG	varchar (1). Beneficiary party charge flag.
BNP_IDTYPE	varchar (1). Beneficiary party ID type.
BNP_ID	varchar (64). Beneficiary party ID.
BNP_BNK_FLG	varchar (1). Flag indicating whether beneficiary party is a bank (Y). or not (N).
BNP_NAME1	varchar (35). Beneficiary party name line 1.
BNP_NAME2	varchar (35). Beneficiary party name line 2.
BNP_NAME3	varchar (35). Beneficiary party name line 3.
BNP_NAME4	varchar (35). Beneficiary party name line 4.
BNP_ADV_INST1	varchar (35). Beneficiary party advice line 1.
BNP_ADV_INST2	varchar (35). Beneficiary party advice line 2.
BNP_ADV_INST3	varchar (35). Beneficiary party advice line 3.
ORP_BEN_INF1	varchar (35). Originator information line 1.
ORP_BEN_INF2	varchar (35). Originator information line 2.
ORP_BEN_INF3	varchar (35). Originator information line 3.
ORP_BEN_INF4	varchar (35). Originator information line 4.
CDT_PAYSYS_FMT	varchar (1). User payment system format indicator.
CDT_STATE	varchar (3). State or province of the credit party.
CDT_POSTAL_CODE	varchar (15). Postal code of the credit party.
CDT_RES_COUNTRY	varchar (2). Resident country of the credit party.
CDT_REF_NUM	varchar (16). Reference number of the credit party.
IBK_STATE	varchar (3). State or province of the intermediary bank party.
IBK_POSTAL_CODE	varchar (15). Postal code of the intermediary bank party.
IBK_RES_COUNTRY	varchar (2). Resident country of the intermediary bank party.
IBK_REF_NUM	varchar (16). Reference number of the intermediary bank party.
IB1_STATE	varchar (3). State or province of the first intermediary bank party.

IB1_POSTAL_CODE	varchar (15). Postal code of the first intermediary bank party.
IB1_RES_COUNTRY	varchar (2). Resident country of the first intermediary bank party.
IB1_REF_NUM	varchar (16). Reference number of the first intermediary bank party.
BBK_STATE	varchar (3). State or province of the beneficiary bank party.
BBK_POSTAL_CODE	varchar (15). Postal code of the beneficiary bank party.
BBK_RES_COUNTRY	varchar (2). Resident country of the beneficiary bank party.
BBK_REF_NUM	varchar (16). Reference number of the beneficiary bank party.
BNP_STATE	varchar (3). State or province of the beneficiary party.
BNP_POSTAL_CODE	varchar (15). Postal code of the beneficiary party.
BNP_RES_COUNTRY	varchar (2). Resident country of the beneficiary party.
BNP_REF_NUM	varchar (16). Reference number of the beneficiary party.
BNP_MAILING_COUNTRY	varchar (2). Mailing country for the beneficiary party.
REGULATORY_REPORT1	varchar (35). First field of regulatory reporting (77B).
REGULATORY_REPORT2	varchar (35). Second field of regulatory reporting (77B).
REGULATORY_REPORT3	varchar (35). Third field of regulatory reporting (77B).
BNP_ADR_BNK_ID	varchar (3). Beneficiary party address bank ID.
BNP_REL_ID	varchar (11). Beneficiary party address reference ID.
CDT_CAMEFROM	varchar (20). Source from where the credit party was found.
IB1_CAMEFROM	varchar (20). Source from where the first intermediary bank party was found.
IBK_CAMEFROM	varchar (20). Source from where the intermediary bank party was found.
BBK_CAMEFROM	varchar (20). Source from where the beneficiary bank party was found.
BNP_CAMEFROM	varchar (20). Source from where the beneficiary party was found.
CREDITSIDE_RESIDENCY	varchar (2). Residency of the ultimate credit party: ORP, OBK, CDT.
CDT_PROFILE_IDBANK	varchar (3). Bank for the credit party profile.
CDT_PROFILE_IDTYPE	varchar (1). ID type for the credit party profile.

Anticipation Template Tables

CDT_PROFILE_IDACC	varchar (30). Account number for the credit party profile.
CDT_PROFILE_IDADR	varchar (5). Address code for the credit party profile.
CDT_PROFILE_IDPAD	varchar (1). Padding for the credit party profile.
CDT_BEI_FLAG	varchar (1). S.W.I.F.T. BIC/BEI flag of the credit party.
IBK_BEI_FLAG	varchar (1). S.W.I.F.T. BIC/BEI flag of the first intermediary bank party.
IB1_BEI_FLAG	varchar (1). S.W.I.F.T. BIC/BEI flag of the first intermediary bank party.
BBK_BEI_FLAG	varchar (1). S.W.I.F.T. BIC/BEI flag of the beneficiary bank party.
BNP_BEI_FLAG	varchar (1). S.W.I.F.T. BIC/BEI flag of the beneficiary party.

Anticipation Template Debit Table

(ANT_TMPL_DR_T)

The Anticipation Template Debit table is a holding place for data from the debit side of anticipation messages.

This table works in conjunction with the ANT_TMPL_T and tables for each anticipation.

Primary Keys

The Anticipation Template Debit table (ANT_TMPL_DR_T) has the following primary keys:

- ANT_TMPL_ID
- ANT_TIME_STAMP

Table Relationships

This table relates to the other anticipation template tables through the ANT_TMPL_ID column.

Column Descriptions

ANT_TMPL_ID (pk)	varchar (20). Anticipation template ID.
ANT_TIME_STAMP (pk)	varchar (16). Record update time.
RECORD_EXPIRED	varchar (16). Date and time when this record becomes invalid.
RECORD_UPDATED	varchar (1). Indicates whether this row has had its timestamp propagated.
DBT_ADR_BNK_ID	varchar (3). Debit party address reference ID.
DBT_REL_ID	number (10). Debit party address reference ID.
DBT_TYP_OVR	<div>varchar (1). Debit party override flag; contains the following values:</div> <ul style="list-style-type: none">• * (Debit party is not in REL)• Space (Debit party is in REL)• (There is an inconsistency in the REL database)

Anticipation Template Tables

DBT_TYP_IDTYPE	<p>varchar (1). Debit party account ID type; mapped from the first part of DBT column on the FTR screen. This column contains the following values:</p> <ul style="list-style-type: none"> • A (Fedwire routing ABA number) • B (Branch code) • C (CHIPS user ID) • D (DDA) • E (Extended ID) • F (Foreign nostro account number) • G (General ledger account number) • I (Interbank lookup key) • K (Customer ID number) • M (Private leased line network MAC code) • N (MTS mnemonic name) • P (CHIPS participant ID) • R (ACI Worldwide system address reference number) • S (8- or 11-character S.W.I.F.T. address) • T (Telex number) • U (User-defined name) • V (Savings account number) • X (Telex answerback) • Z (Cable address)
DBT_TYP_ID	varchar (64). Debit party account ID.
DR_ACC_BANK_ID	varchar (3). A differentiator to allow configurable cross bank indices.
DR_ACC_ID_TYPE	<p>varchar (1). Debit account ID type; contains the following values:</p> <ul style="list-style-type: none"> • D (DDA) • F (Foreign nostro account number) • G (General ledger account number) • P (CHIPS participant ID) • V (Savings account number)
DR_ACC_ACCOUNT	varchar (30). Account number.
DBT_AMOUNT	(number 21.3). Debit amount.
DBT_CURRENCY	varchar (3). Currency of the debit party in the anticipation.
SEC_FLG	varchar (16). Secondary account flag.

SEC_AMT	number (21.3). Secondary amount, either charges or commissions.
SEC_CUR	varchar (3). Secondary currency.
SEC_ACC_BANK_ID	varchar (3). A differentiator to allow configurable cross bank indices.
SEC_ACC_ID_TYPE	varchar (1). Secondary account ID type; contains the following values: <ul style="list-style-type: none"> • D (DDA) • F (Foreign nostro account number) • G (General ledger account number) • P (CHIPS participant ID) • V (Savings account number)
SEC_ACC_ACCOUNT	varchar (30). Account number that identifies the secondary account.
TER_FLG	varchar (16). Tertiary account flag.
TER_AMT	number 21.3). Tertiary amount, either charges or commissions.
TER_CUR	varchar (3). Tertiary currency.
TER_ACC_BANK_ID	varchar (3). A differentiator to allow configurable cross bank indices.
TER_ACC_ID_TYPE	varchar (1). Tertiary account ID type; contains the following values: <ul style="list-style-type: none"> • D (DDA) • F (Foreign nostro account number) • G (General ledger account number) • P (CHIPS participant ID) • V (Savings account number)
TER_ACC_ACCOUNT	varchar (30). Account number that identifies the tertiary account.
DBT_ACC_CLASS	varchar (6). Debit account class.
DBT_ACC_CITY	varchar (15). Debit account city.
DBT_ACC-COUNTRY	varchar (2). Debit account country.
DBT_ACC_PARENT	varchar (10). Debit account parent code.
DBT_ACC_PRODS	varchar (20). Debit account product codes.
DBT_DEPARTMENT	varchar (20). Debit department.

Anticipation Template Tables

DBT_FEE_CODE	varchar (8). Debit fee code.
DBT_SEC_SUB_ACCT	varchar (61). Securities sub account.
DBT_SYS_OF_REC	varchar (3). Indicates another system from which the balance authorization came.
CONCEN_BANK_ID	varchar (3). A differentiator to allow configurable cross bank indices.
CONCEN_ID_TYPE	varchar (1). Concentration account ID type; contains the following: <ul style="list-style-type: none"> • D (DDA) • F (Foreign nostro account number) • G (General ledger account number) • P (CHIPS participant ID) • V (Savings account number)
CONCEN_ACCOUNT	varchar (30). Concentration account number.
DBT_PARENT_CODE	varchar (10). Debit account parent code.
DBT_ADR_TYPE	varchar (6). Debit address type.
DBT_ADR_CLASS	varchar (6). Debit address class.
DBT_ADR_CITY	varchar (15). Debit address city.
DBT_ADR_COUNTRY	varchar (2). Debit address country.
DBT_ADV_TYP	varchar (3). Debit advice type.
DBT_RT_CODE	varchar (6). Debit routing code.
DBT_RT_STATE	varchar (1). Indicates whether the routing code instructions have been queued. Q = routing code instructions are queued
DR_WIR_IDTYPE	varchar (1). ID type for party (address) indices.
DR_WIR_IDKEY	varchar (64). Name, or routing ID, or account number/address code.
DBT_NAME1	varchar (25). Debit name line 1.
DBT_NAME_2	varchar (35). Debit name line 2.
DBT_NAME_3	varchar (35). Debit name line 3.
DBT_NAME_4	varchar (35). Debit name line 4.

DBT_DRAFT_NUM	varchar(16). Debit draft number.
DBT_RECON_REF	varchar (12). G/L reconciliation reference number.
DBT_SPC_INST1	varchar (35). Debit special instructions line 1.
DBT_SPC_INST2	varchar (35). Debit special instructions line 2.
DBT_SPC_INST3	varchar (35). Debit special instructions line 3.
DBT_HOLD	varchar (1). Account hold flag.
DBT_BAL	varchar (1). Account balance exception flag.
DBT_LIM	varchar (1). Account limit exception flag.
CHIPS_LIM	varchar (1). CHIPS transfer limit exception flag.
DBT_GRP_BAL	varchar (1). Group balance exception flag.
DBT_DRAWDOWN	varchar (1). Account drawdown exception flag.
DBT_PRE	varchar (1). Account preadvised exception flag.
DBT_GRP_PRE	varchar (1). Group preadvised exception flag.
DBT_PS_ELIG	<p>varchar (1). Flag indicating whether the debit party is eligible for pre or split advising. This column contains the following values:</p> <ul style="list-style-type: none"> • S (Eligible for split-date accounting) • P (Eligible for preadvising) • N (Eligible for send-date accounting, not preadvising) • Blank (Not relevant)
DBT_ITEM_HOLD	varchar (1). Value O indicates the risk item hold is overridden.
DBT_NOF_LOOKED_UP	varchar (1). Value T indicates not-on-file debit part has been looked up.
DDA_BALANCE	varchar (9). DDA balance.
BALANCE_RISK	varchar (8). Balance risk.
SBK_ADR_BNK_ID	varchar (3). Sending bank party address bank ID.
SBK_REL_ID	number (10). Sending bank party address reference ID.
SBK_IDTYPE	varchar (1). Sending bank ID type.
SBK_ID	varchar (64). Sending bank ID.

Anticipation Template Tables

NAME1	varchar (35). Sending bank line 1.
NAME2	varchar (35). Sending bank line 2.
NAME3	varchar (35). Sending bank line 3.
NAME4	varchar (35). Sending bank line 4.
REF_NUM	varchar (16). Sending bank reference number.
OBK_ADR_BNK_ID	varchar (3). Ordering bank address ID.
OBK_REL_ID	number (10). Ordering bank relationship ID.
OBK_IDTYPE	varchar (1). Ordering bank ID type.
OBK_ID	varchar (64). Ordering bank ID.
OBK_NAME1	varchar (35). Ordering bank name line 1.
OBK_NAME2	varchar (35). Ordering bank name line 2.
OBK_NAME3	varchar (35). Ordering bank name line 3.
OBK_NAME4	varchar (35). Ordering bank name line 4.
OBK_REF_NUM	varchar (16). Ordering bank reference number.
ORP_ADR_BNK_ID	varchar (3). Originator bank address ID.
ORP_REL_ID	number (10). Originator relationship ID.
ORP_IDTYPE	varchar (1). Originator ID type.
ORP_ID	varchar (64). Originator ID.
ORP_NAME1	varchar (35). Originator name line 1.
ORP_NAME2	varchar (35). Originator name line 2.
ORP_NAME3	varchar (35). Originator name line 3.
ORP_NAME4	varchar (35). Originator name line 4.
ORP_REF_NUM	varchar (16). Originator reference number.
DBT_BNK_INF1	varchar (35). Debit bank information line 1.
DBT_BNK_INF2	varchar (35). Debit bank information line 2.

DBT_BNK_INF3	varchar (35). Debit bank information line 3.
DBT_BNK_INF4	varchar (35). Debit bank information line 4.
DBT_BNK_INF5	varchar (35). Debit bank information line 5.
DBT_BNK_INF6	varchar (35). Debit bank information line 6.
INS_ADR_BNK_ID	varchar (3). Instructing bank ID.
INS_REL_ID	number (10). Instructing bank REL ID.
INS_IDTYPE	varchar (1). Instructing bank ID type.
INS_ID	varchar (64). Instructing bank ID.
INS_NAME1	varchar (35). Instructing bank name line 1.
INS_NAME2	varchar (35). Instructing bank name line 2.
INS_NAME3	varchar (35). Instructing bank name line 3.
INS_NAME4	varchar (35). Instructing bank name line 4.
INS_REF_NUM	varchar (16). Instructing bank reference number.
RCA_ADR_BNK_ID	varchar (3). Receiving agent bank ID.
RCA_REL_ID	number (10). Receiving agent REL ID.
RCA_IDTYPE	varchar (1). Receiving agent ID type.
RCA_ID	varchar (64). Receiving agent ID.
RCA_NAME1	varchar (35). Receiving agent name line 1.
RCA_NAME2	varchar (35). Receiving agent name line 2.
RCA_NAME3	varchar (35). Receiving agent name line 3.
RCA_NAME4	varchar (35). Receiving agent name line 4.
RCA_REF_NUM	varchar (16). Receiving agent reference number.
DBT_STATE	varchar (3). State or province of the debit party.
DBT_POSTAL_CODE	varchar (15). Postal code of the debit party.
DBT_RES_COUNTRY	varchar (2). Resident country of the debit party.

Anticipation Template Tables

DBT_REF_NUM	varchar (16). Reference number of the debit party.
SBK_STATE	varchar (3). State or province of the sending bank party.
SBK_POSTAL_CODE	varchar (15). Postal code of the sending bank party.
SBK_RES_COUNTRY	varchar (2). Resident country of the sending bank party.
OBK_STATE	varchar (3). State or province of the ordering bank party.
OBK_POSTAL_CODE	varchar (15). Postal code of the ordering bank party.
OBK_RES_COUNTRY	varchar (2). Resident country of the ordering bank party.
ORP_STATE	varchar (3). State or province of the originating party.
ORP_POSTAL_CODE	varchar (15). Postal code of the originating party.
ORP_RES_COUNTRY	varchar (2). Resident country of the originating party.
INS_STATE	varchar (3). State or province of the instructing bank party.
INS_POSTAL_CODE	varchar (15). Postal code of the instructing bank party.
INS_RES_COUNTRY	varchar (2). Resident country of the instructing bank party.
RCA_STATE	varchar (3). State or province of the receiving agent.
RCA_POSTAL_CODE	varchar (15). Postal code of the receiving agent.
RCA_RES_COUNTRY	varchar (2). Resident country of the receiving agent.
DBT_CAMEFROM	varchar (20). Source where the debit party was found.
SBK_CAMEFROM	varchar (20). Source where the sending bank party was found.
OBK_CAMEFROM	varchar (20). Source where the ordering bank party was found.
OPR_CAMEFROM	varchar (20). Source where the originating party was found.
INS_CAMEFROM	varchar (20). Source where the instructing party was found.
RCA_CAMEFROM	varchar (20). Source where the receiving agent was found.
DEBITSIDE_RESIDENCY	varchar (2). Residence of the ultimate debit party: <ul style="list-style-type: none"> • ORP = Originating party • OBK = Originating bank • DBT = Debit party

DBT_PROFILE_IDBANK	varchar (3). ID of the Bank for the debit party profile.
DBT_PROFILE_IDTYPE	varchar (1). ID type of the debit party profile.
DBT_PROFILE_IDACC	varchar (30). Account number for the debit party profile.
DBT_PROFILE_IDADR	varchar (5). Address code for the debit party profile.
DBT_PROFILE_IDPAD	varchar (1). Padding for the debit party profile.
SBK_PROFILE_IDBANK	varchar (3). ID of the sending bank party profile.
SBK_PROFILE_IDTYPE	varchar (1). ID type for the sending bank party profile.
SBK_PROFILE_IDACC	varchar (30). Account number for the sending bank profile.
SBK_PROFILE_IDADR	varchar (5). Address code for the sending bank profile.
SBK_PROFILE_IDPAD	varchar (1). Padding for the sending bank profile.
DBT_BEI_FLAG	varchar (1). S.W.I.F.T. BIC/BEI flag of the debit party.
SBK_BEI_FLAG	varchar (1). S.W.I.F.T. BIC/BEI flag of the sending bank party.
OBK_BEI_FLAG	varchar (1). S.W.I.F.T. BIC/BEI flag of the ordering bank party.
ORP_BEI_FLAG	varchar (1). S.W.I.F.T. BIC/BEI flag of the originating party.
INS_BEI_FLAG	varchar (1). S.W.I.F.T. BIC/BEI flag of the instructing bank party.
RCA_BEI_FLAG	varchar (1). S.W.I.F.T. BIC/BEI flag of the receiving agent party.

Bank Tables

The Bank tables contain all the information for the banks in the system, including foreign currency tolerances, special account information, routing information, risk operation, and fees.

Parent/child relationships

The Bank table (BANK_T) is the parent table to the Fee table.

Primary keys

The Bank tables have the following primary keys in common:

- BANK_ID
- BNK_TIME_STAMP

Bank Table (BANK_T)

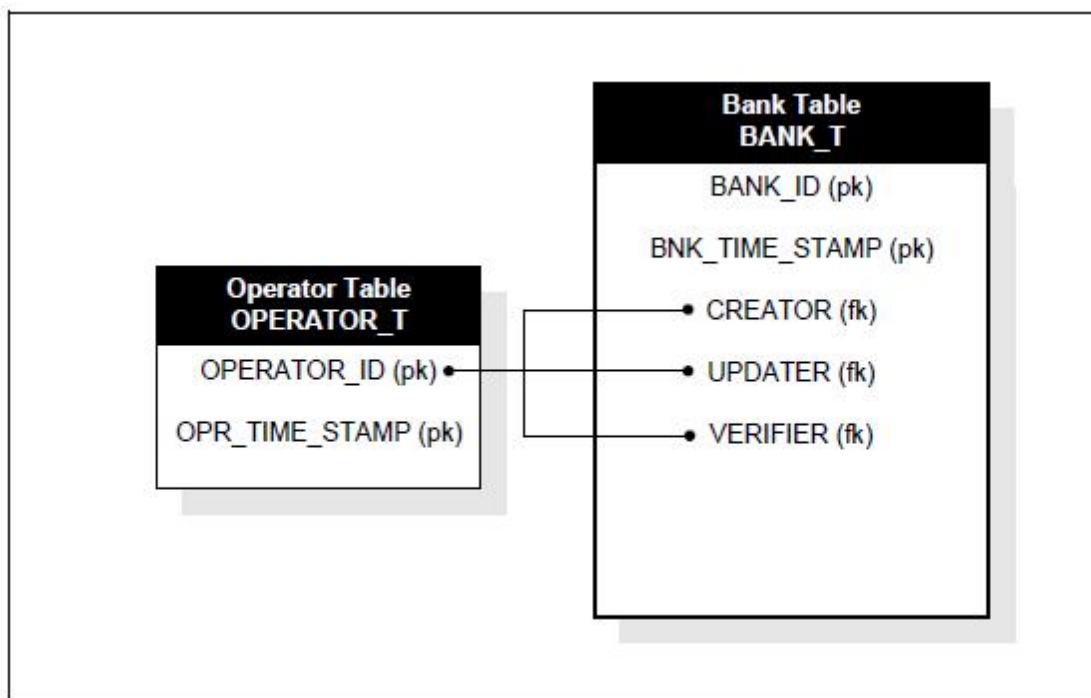
The Bank table contains bank-specific information including the bank's ABA number, CHIPS UIDs, and risk and routing parameters.

Table Relationships

This section lists the keys in the Bank table that relate to keys in other tables. In addition to the relationships listed here, the BANK_ID column relates to foreign keys in most of the other database tables. Please refer to the relationship diagrams in other tables to see what foreign keys in those tables relate to the BANK_ID column in the Bank table.

Relationship Diagram

This relationship diagram shows the Bank table along with the Operator table.



Column Descriptions

BANK_ID (pk)	(varchar 3). Bank identifier.
BNK_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.

Bank Tables

DAT_FUNCTION	(varchar 7). Function that last accessed this bank record; contains the following values: <ul style="list-style-type: none">• VFY ADD (Verify add)• VFY DEL (Verify delete)• VFY UPD (Verify update)
CREATOR (fk)	(varchar 10). Operator ID and bank ID of the operator who created the bank record.
CREATE_DATE	(date). Date when the bank record was created.
UPDATER (fk).	(varchar 10). Operator ID and bank ID of the operator who updated the bank record.
UPDATE_DATE	(date). Date when the bank record was updated.
VERIFIER (fk)	(varchar 10). Operator ID and bank ID of the operator who verified the bank record.
VERIFY_DATE	(date). Date when the bank record was verified.
DAT_OWNER_PROD_ID	(varchar 3). Product ID.
DAT_OWNER_BANK_ID	(varchar 3). Bank ID.
DAT_OWNER_LOC	(varchar 6). Location ID.
DAT_OWNER_CUST	(varchar 20). Customer ID.
BANK_NAME	(varchar 35). Bank name.
SYS_NAME	(varchar 35). Bank-specified system name for the Money Transfer System.
BASE_CUR_ID	(varchar 3). ISO currency code for the base currency.
BASE_CUR_PREC	(varchar 1). Number of decimal places allowed for the base currency.
ABA_NUMBER	(varchar 9). American Bankers Association Routing and Transit number.
FED_NAME	(varchar 20). Bank's short name at the Federal Reserve.
CHIPS_ID	(varchar 4). CHIPS identifier.
SWIFT_ID	(varchar 30). S.W.I.F.T. identifier.
TELEX_ANS	(varchar 20). Telex answerback.
LOGIN_TRYS	(number 5). Number of login tries permitted before a user is suspended.

PSWRD_GOOD_DAYS	(number 5). Number of days that a password remains valid.
BUSY_TIMER_MINS	(varchar 2). Number of minutes messages are held on the Busy queue for Verify Callback processing.
HIGH_PRIO_AMT	(number 21.3). High priority amount.
PHONE_AUT_AMT	(number 21.3). Phone authorization amount.
CHECK_HIGH_AMT	(number 21.3). Check high amount.
CHECK_LOW_AMT	(number 21.3). Check low amount.
FF_BUY_PCT	(number 18.3). Not populated in this table.
FF_SELL_PCT	(number 18.3). Not populated in this table.
FF_ROLLOVER_PCT	(number 18.3). Not populated in this table.
FF_TOLERANCE_AMT	(number 21.3). Not populated in this table.
FF_LOAN_DAYS_REG	(number 5). Number of days until maturity for the bank's Fed Fund trades.
FX_APPROVAL_AMT	(number 21.3). Foreign exchange approval amount; any amount above this amount (in base currency). needs a trader control number.
FX_TOLERANCE_PCT	(number 18.3). Foreign exchange conversion tolerance that determines how much a non-locked rate can deviate from a standard exchange rate.
HIGH_VFY_AMT	(number 21.3). Secondary verification amount; any amount above this amount requires secondary verification.
COMMISSION	(number 21.3). Reserved for future use.
PRI_PEAK_BEG_HRS	(varchar 2). Hour portion of primary peak period start time.
PRI_PEAK_BEG_MNS	(varchar 2). Minute portion of primary peak period start time.
PRI_PEAK_END_HRS	(varchar 2). Hour portion of primary peak period end time.
PRI_PEAK_END_MNS	(varchar 2). Minute portion of primary peak period end time.
SEC_PEAK_BEG_HRS	(varchar 2). Hour portion of secondary peak period start time.
SEC_PEAK_BEG_MNS	(varchar 2). Minute portion of secondary peak period start time.
SEC_PEAK_END_HRS	(varchar 2). Hour portion of secondary peak period end time.
SEC_PEAK_END_MNS	(varchar 2). Minute portion of secondary peak period end time.
PREF_FED_BNK_ID	(varchar 3). Bank through which FED payments are made.

Bank Tables

PREF_CHP_BNK_ID	(varchar 3). Bank through which CHIPs payments are made.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
PSWRD_REUSE_DAYS	(number 5). Number of days before an operator's password can be used again.
COUNTRY_CODE	(varchar 2). Country code of the bank used to determine the cross-country payments.
FEE_MASTER_BNK	(varchar 3). The bank whose fee table is to be used.
PHASE_NAME	(varchar 10). Name of the phase in which the bank operates.
SEQUENCE_NUMBER_OFFSET	(number 10). Sequence number offset for TRNs in this phase for this bank.
CURRENT_NUMBER_OFFSET	(number10). Current serial register value for TRNs in this phase in this bank.

Bank Clearinghouse Table (BANK_CLEARHOUSE_T)

The Bank Clearinghouse table contains data of banks that act as clearinghouses, a clearinghouse being an entity that processes transactions. Information in the table for bank clearinghouses includes what kind of transactions the bank can process and when.

Primary Keys

The Bank Clearinghouse table (BANK_CLEARHOUSE_T) has the following primary keys:

- BANK_ID
- BANK_CLRHS_ADV_TYPE
- BNK_TIME_STAMP

Column Descriptions

BANK_ID (pk)	(varchar 3). Bank identifier.
BANK_CLRHS_ADV_TYPE (pk)	(varchar 3). Bank clearinghouse advice type.
BNK_TIME_STAMP (pk)	(number 16). Time at which the record was updated.
BNK_CLRHS_SWF_ACC_TYPE	(varchar 2). Bank clearinghouse SWIFT account type.
BNK_CLRHS_CURRENCY	(varchar 3). Bank clearinghouse ISO currency code.
BNK_CLRHS_FORMATTER	(varchar 3). Bank clearinghouse format type.
BNK_CLRHS_MIN_LENGTH	(number 5). Bank clearinghouse minimum length.
BNK_CLRHS_MAX_LENGTH	(number 5). Bank clearinghouse maximum length.
BNK_CLRHS_MOD_CHECK	(varchar 3). Bank clearinghouse modulus check.
BNK_CLRHS_CH_NAME	(varchar 9). Bank clearinghouse accounting name.
BNK_CLRHS_PID_SUFFIX	(varchar 3). Suffix that differentiates clearinghouses.
BNK_CLRHS_CLR_SYS	(varchar 3). Clearing system name for the LQM queues.
BNK_CLRHS_DESTYPE	(varchar 3). Bank clearinghouse destination type.

Bank Tables

BNK_CLRHS_DFINCPY	(varchar 3). Bank clearinghouse special destination sub-type.
BNK_CLRHS_LQM_FNC_NAME	(varchar 3). Liquidity management function name.
BNK_CLRHS_CLR_SYS_TYPE	(varchar 2). Type of settlement. RT = RTGS (real time gross settlement) NT = Netting CB = Correspondent
BNK_CLRHS_CLR_SYS_BIC	(varchar 11). Identifier code of the central bank or clearhouse switch.
BNK_CLRHS_CLR_SYS_GL	(varchar 30). General ledger account number for payments made through this central bank. This may be a Nostro account number.
BNK_CLRHS_CLR_SYS_NSTRO	(varchar 30). Account number of the central bank's account.
BNK_CLRHS_CLR_SYS_GLTYPE	(char 1). ID type of the general ledger account number.
RECORD EXPIRED	(number 16). Date and time when this record becomes invalid.
RECORD UPDATED	(char 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Bank Special Table (BANK_SPECIAL_T)

The Bank Special table provides detailed information about special accounts.

Some examples of special accounts are: Fed G/L, Fed Funds principal, Fed Funds interest, multibank operations, or cross-bank accounts.

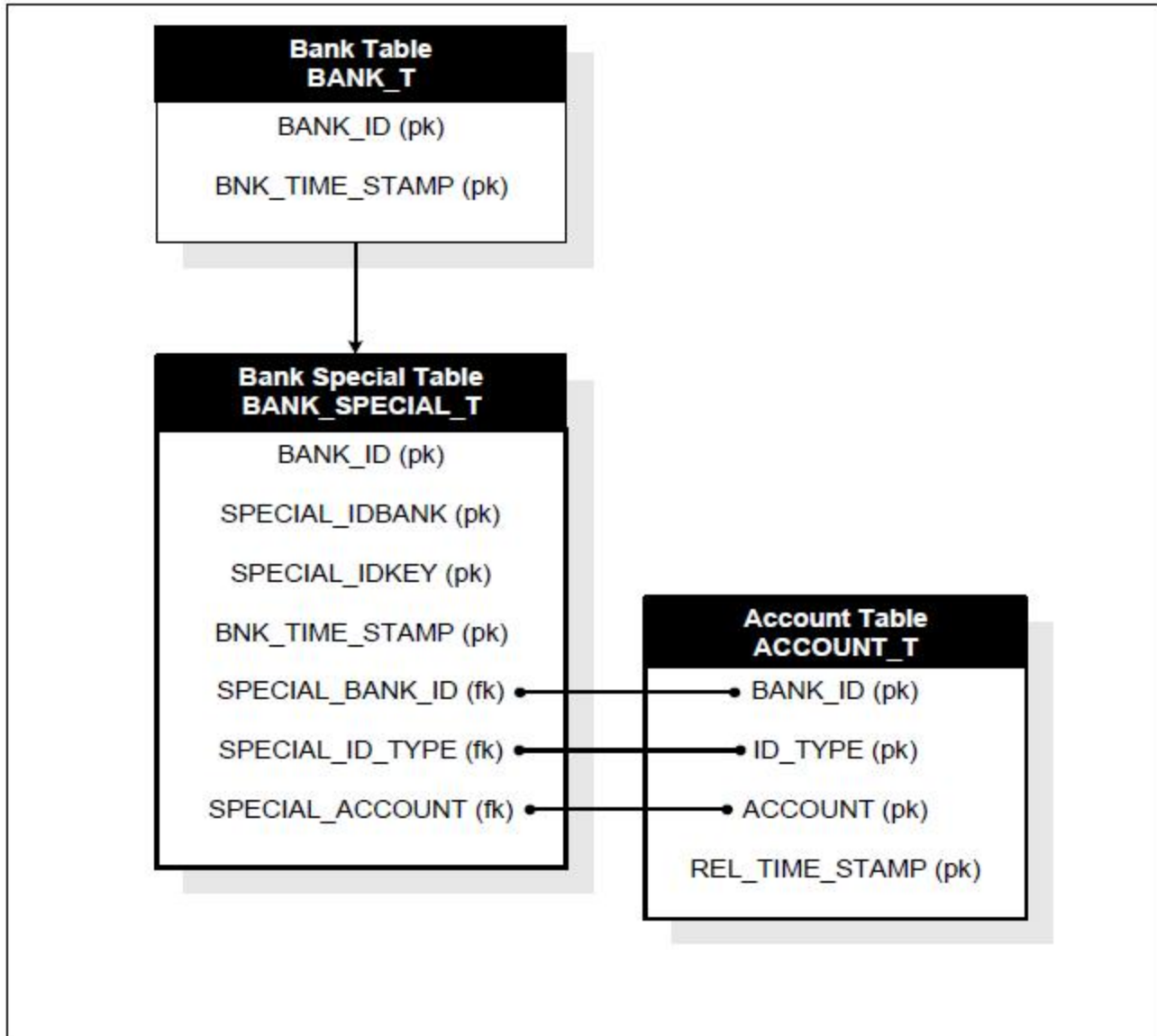
The following section lists the keys in the Bank Special table that relate to keys in other tables.

Account table

The SPECIAL_BANK_ID, SPECIAL_ID_TYPE, and SPECIAL_ACCOUNT columns in the Bank Special table relate to the BANK_ID, ID_TYPE, and ACCOUNT columns in the Account table.

Relationship Diagram

This relationship diagram shows the Bank Special table along with its parent table, the Bank table (BANK_T) and the Account table.



Column Descriptions

BANK_ID (pk)	(varchar 3). Bank identifier.
SPECIAL_IDBANK (pk)	(varchar 3). Bank-specific ID for the special account's owning bank. A special account can be one of the following types of accounts: Fed G/L, CHIPS, Fed Funds, Fed Funds interest, multibank operations, or cross-bank accounting.

SPECIAL_IDKEY (pk)	(varchar 20). Bank-specific name used to identify a special account.
BNK_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
SPECIAL_BANK_ID (fk)	(varchar 3). Bank that owns the special account record; same as the BANK_ID column.
SPECIAL_ID_TYPE (fk)	(varchar 1). Account ID type of the special account; contains the following values: <ul style="list-style-type: none">• D (DDA)• F (Foreign nostro account number)• G (General ledger account number)• P (CHIPS participant ID)• V (Savings account number)
SPECIAL_ACCOUNT (fk)	(varchar 30). Special account ID.
INDIRECT_IDBANK	(varchar 3). Bank-specific ID for the indirect account's owning bank.
INDIRECT_IDKEY	(varchar 20). Bank-specific name used to identify an indirect account.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Bank Location Cost Table (BANK_LOC_COST_T)

The Bank Location Cost table provides information on linking bank location and cost center combinations to a particular bank.

Primary Keys

The Bank Location Cost table (BANK_LOC_COST_T) has the following primary keys:

- BNK_ID
- BNK_LOC
- TPL_TIME_STAMP

Column Descriptions

BANK_ID (pk)	(varchar 3). Bank identifier.
BNK_LOC (pk)	(varchar 6). Bank's location.
BNK_TIME_STAMP (pk)	(varchar 16). Date and time when the bank location/cost center record was updated.
BNK_COST_CTR	(varchar 20). Bank's cost center.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Bank Fee Table (FEE_T)

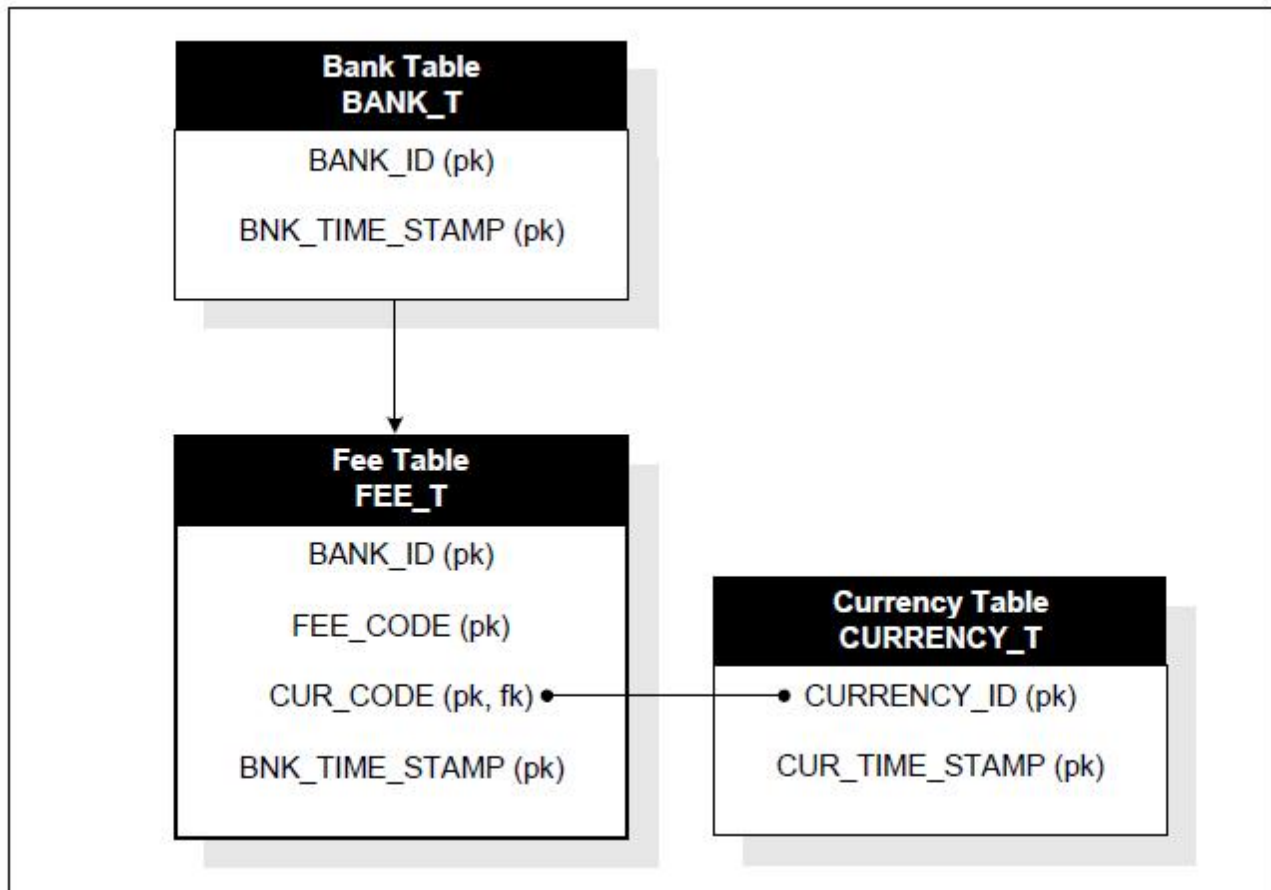
The Bank Fee table associates bank charges and costs to the fee codes in the fee matrix.

Table Relationships

The CUR_CODE column in the Bank Fee table relates to the CURRENCY_ID column in the Currency table.

Relationship Diagram

This relationship diagram shows the Bank Fee table along with its parent table, the Bank table, and the Currency table.



Column Descriptions

BANK_ID (pk)	(varchar 3). Bank identifier of the bank that owns this account; relates to the BANK field on the FEE screen.
FEE_CODE (pk)	(varchar 8). Fee code used to identify a fee; relates to the FEE field on the FEE screen.
CUR_CODE (pk, fk)	(varchar 3). Currency code; relates to the CUR field on the FEE screen.
BNK_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
CHARGE	(number 21.3). Amount to charge a customer for the transaction.
COST	(number 21.3). Bank's cost to process the transaction (may include overhead or wire service costs). .
DISCOUNT_CNT	(number 10). A threshold count on the number of transactions after which a bank will offer discounts.
DISCOUNT_AMT	(number 21.3). Discount amount.
DISCOUNT_PCT	(number 18.16). Discount percentage.
FEE_CAT	(varchar 20). Fee category; relates to the CATEGORY field on the FEE screen.
FEE_DESC	(varchar 20). Fee description; relates to the DESCRIPTION field on the FEE screen.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Change Tables

The RGW Change tables work with the File Change Data Loader to provide RGW with the information necessary to generate file change reports.

Primary Keys

The change tables share the following primary key:

- CHANGE_TABLE

Change List Table (CHANGE_TBL_LIST_T)

The Change List table stores data that enables the File Change Data Loader to: identify the table names and primary keys of those tables whose data appears in the file change reports, and to determine the most recent version of records in those tables.

Primary Keys

The Change List table (CHANGE_TBL_LIST_T) has the following primary key:

- CHANGE_TABLE

Column Descriptions

CHANGE_TABLE (pk)	(varchar 30). Identifies the name of the table to be loaded into the change file data table.
TIMESTAMP_FIELD	(varchar 18). Name of the timestamp field for the table in the Change_Table field.
EXPIRED_FIELD	(varchar 18). Name of the expiration timestamp field for the table in the Change_Table field.
LAST_TIMESTAMP	(varchar 20). Last timestamp value for the table in the Change_Table field.
LAST_EXPIRED	(varchar 20). Expiration timestamp value for the table in the Change_Table field.
BANK_FIELD	(varchar 18). Name of the field in other tables that contain the Bank ID.
ID_TYPE_FIELD	(varchar 18). Name of the field in other tables that contains the ID type.
ID_FIELD	(varchar 60). Name of the field in other tables that contains the ID, for example, account ID.
ID2_FIELD	(varchar 60). Name of the field in other tables that contains the second ID, for example, repetitive ID, stop number, and others.
ID3_FIELD	(varchar 60). Name of the field in other tables that contains the third ID, for example, standing order ID.
OBJECT_ID_FIELD	(varchar 18). Name of the field in other tables that contains the object ID.
OPERATOR_ID_FIELD	(varchar 18). Name of the field in other tables that contains the operator ID.
VERIFIER_ID_FIELD	(varchar 18). Name of the field in other tables that contains the verifier ID
RECORD_EXPIRED	(varchar 16). Date and time that this record became invalid.

Change Data Table (CHANGE_DATA_T)

The Change Data table stores old and new values for fields specified in the Change List table.

Each time you run the File Change Data Loader, this table maintains a record of the changed fields. For example, if you run the File Change Data Loader on June 8th and June 9th, the Change Data Table stores changed fields for both dates.

Note:

This table does not store changed data for fields in the Change Exclusion table.

Primary Keys

The Change Data table (CHANGE_DATA_T) has the following primary keys:

- CHANGE_DATE
- PRODUCT
- BANK_ID
- ID_TYPE
- ID
- ID2
- ID3
- OBJECT_ID
- CHANGE_TABLE
- CHANGE_COLUMN

Column Descriptions

CHANGE_DATE (pk)	(varchar 16). Date and time that this record last changed.
PRODUCT (pk)	(varchar 3). Product identification number.
BANK_ID (pk)	(varchar 3). Bank identification number.
ID_TYPE (pk)	(varchar 3). Repetitive ID type or relationship ID type.

Change Tables

ID (pk)	(varchar 60). Account ID, PIN ID, or STOP source operator ID as needed.
ID2 (pk)	(varchar 60). Repetitive ID or the STOP number.
ID3 (pk)	(varchar 60). Standing Orders ID, STOP token, or STOP name (as needed).
OBJECT_ID (pk)	(varchar 32). ID from the MTS transaction object store.
CHANGE_TABLE (pk)	(varchar 30). Table name that contains this data.
CHANGE_COLUMN (pk)	(varchar 30). Name of the column that was changed.
OLD_CONTENTS	(varchar 300). Value of this field prior to the most recent update.
NEW_CONTENTS	(varchar 300). Value of this field after the most recent update.
DATA_TYPE	(varchar 106). Original data type for the field that has changed.
OPERATOR_ID	(varchar 10). Identification number of the operator who updated this record.
VERIFIER_ID	(varchar 10). Identification number of the operator who verified this record.
OPERATION	(varchar 1). Indicates how the record was changed. Valid values include: <ul style="list-style-type: none">• A (Add)• U (Update)• D (Delete)
RECORD_EXPIRED	(varchar 16). Date and time that the record becomes invalid.

Change Exclusion Table (CHANGE_TBL_EXC_T)

The Change Exclusion table identifies fields not stored in the Change Data table.

Primary Keys

The Change Exclusion table (CHANGE_TBL_EXC_T) has the following primary keys:

- CHANGE_TABLE
- CHANGE_COLUMN

Column Descriptions

CHANGE_TABLE (pk)	(varchar 30). Identifies the name of the table to be loaded into the change file data table.
CHANGE_COLUMN (pk)	(varchar 30). Identifies the name of the column to exclude from change tracking.
RECORD_EXPIRED	(varchar 16). Date and time when this record became invalid.

Channel Tables

Channel tables contain information about channel relationships such as CHIPS, FED, or clearing house.

Channels are relationships that define where information from MTS is to be forwarded. The destination is very often CHIPS, Fed, or clearing house, but can be any system that receives data from MTS. A channel file set up in REL consists of a set of processing rules. The information in a message is compared to the criteria in these processing rules to determine whether the message can be sent through this channel. For example, you can set time or amount limits for messages going through a particular channel.

Primary keys

The Channel tables have these primary keys in common:

- CHANNEL_ID
- CHANNEL_TIME_STAMP

Channel Table (CHANNEL_T)

The Channel table contains control information for either a source or a distribution delivery destination for a transaction.

The channel is a relationship that defines where information from MTS is to be forwarded. The destination is very often CHIPS, Fed, or clearing house, but can be any system that receives data from MTS. The source can be fax, Telex, or SWIFT, for example.

Primary Keys

The Channel table (CHANNEL_T) has the following primary keys:

- CHANNEL_ID
- BANK_ID
- CHANNEL_TIME_STAMP

Column Descriptions

CHANNEL_ID (pk)	(varchar 24). Channel ID.
CHANNEL_TIME_STAMP (pk)	(number 16). Date and time when the channel record was updated.
REL_ID	(number 10). Unique reference number that identifies this relationship.
BANK_ID (pk)	(varchar 3). ID of the bank. This makes the bank unique in a multibank environment.
DAT_FUNCTION	(varchar 7). Function that last accessed this account record; contains the following values: <ul style="list-style-type: none">• VFY ADD (Verify add)• VFY DEL (Verify delete)• VFY UPD (Verify update)
CREATOR	(varchar 10). Operator ID and bank ID of the operator who created the record.
CREATE_DATE	(date). Date when the record was created.
UPDATER	(varchar 10). Operator ID and bank ID of the operator who last updated the record.
UPDATE_DATE	(date). Date on which the record was updated.

Channel Tables

VERIFIER	(varchar 10). Operator ID and bank ID of the operator who verified the record.
VERIFY_DATE	(date). Date on which the record was verified.
DAT_OWNER_PROD_ID	(varchar 3). Product ID.
DAT_OWNER_BANK_ID	(varchar 3). Bank ID.
DAT_OWNER_LOC	(varchar 6). Location ID.
DAT_OWNER_CUST	(varchar 20). Customer ID.
CHANNEL_NAME	(varchar 35). Channel Name.
CHANNEL_DESC	(varchar 35). Description of the channel.
DOCUMENT_ID	(varchar 6). Document ID.
COUNTRY_CODE	(char 2). ISO code that identifies the country.
CHANNEL_TYPE	(varchar 20). Channel type.
CHANNEL_ACCT_BANK_ID	(varchar 3). Differentiator to allow configurable cross bank indices.
CHANNEL_ACCT_ID_TYPE	(varchar 1). ID type for the account indices. Can be one of the following: <ul style="list-style-type: none"> • D = DDA • G = General Ledger • F = Foreign Nostro • P = CHIPS Participant ID • V = Vostro
CHANNEL_ACCT	(varchar 30). Account number.
CHANNEL_LQM_Q_PROD_ID	(varchar 3). Product ID.
CHANNEL_LQM_Q_BANK_ID	(varchar 3). Bank ID.
CHANNEL_LQM_Q_LOC	(varchar 6). Location ID.
CHANNEL_LQM_Q_CUST	(varchar 20). Customer ID.
CHANNEL_LQM_Q_NAME	(varchar 33). Name.
CHANNEL_FORMAT_RTN	(varchar 80). Name of the special formatting routine.
CHANNEL_POSTFRMT_RTN	(varchar 80). Name of the special post-format routine.

CHANNEL_EDIT_RTN	(varchar 80). Name of the special edit routine.
CHANNEL_INBOUND_RTN	(varchar 80). Name of the handler for inbound messages.
CHANNEL_ADV_DEST_RTN	(varchar 80). Name of the handler for the advice destination.
CHANNEL_STATUS	(varchar 80). Channel status.
RECORD_EXPIRED	(number 16). Date and time that this record became invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Channel Address Type Table (CHANNEL_ADR_TYP_T)

The Channel Address Type table contains additional channel address types, and their descriptions, that can be used by the clearing channel. The table identifies, by clearing channel, what kind of IDs can be processed.

Primary Keys

The Channel Address Type table (CHANNEL_ADR_TYP_T) has the following primary keys:

- CHANNEL_ID
- BANK_ID
- CHANNEL_ID_PREFERENCE
- CHANNEL_TIME_STAMP

Column Descriptions

CHANNEL_ID (pk)	(varchar 24). Channel ID.
BANK_ID (pk)	(varchar 3). Bank ID. This makes the bank unique in a multibank environment.
CHANNEL_ID_PREFERENCE (pk)	(number 10). Unique number indicating the preference of this ID.
CHANNEL_TIME_STAMP (pk)	(number 16). Date and time when this record was updated.
CHANNEL_EXT_ID_IDTYPE	(char 1). ID type.
CHANNEL_EXT_ID_SUBTYPE	(varchar 2). ID subtype.
CHANNEL_EXT_ID_TAG	(varchar 6). Tag used for matching an address-associated ID.
RECORD_EXPIRED	(number 16). Date and time that this record became invalid.
RECORD_UPDATED	(char 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Configuration Tables

The Configuration tables contain all configuration information for your system.

Parent/child relationships

The Configuration table (CONFIG_T) is the parent table to the following tables:

- Configuration Item table (CONFIG_ITEM_T)
- Configuration Item Data table (CONFIG_ITEM_DATA_T)
- Configuration Item Description table (CONFIG_ITEM_DESC_T)

Primary keys

The Configuration tables have these primary keys in common:

- NAME
- CFG_TIME_STAMP

Configuration Table (CONFIG_T)

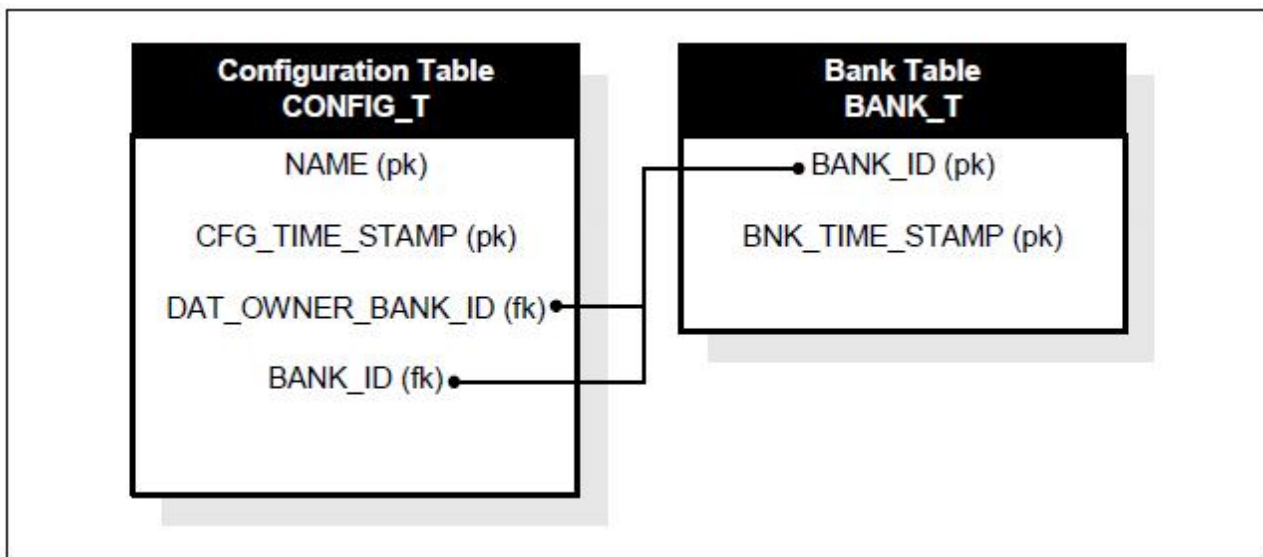
The Configuration table shows all categories of tables in your system.

Table Relationships

The DAT_OWNER_BANK_ID and BANK_ID columns in the Configuration table relate to the BANK_ID column in the Bank table.

Relationship Diagram

This relationship diagram shows the Configuration table along with the Bank table.



Column Descriptions

NAME (pk)	(varchar 33). Name of the table or other configuration data being described.
CFG_TIME_STAMP (pk)	(varchar 16). Time at which the record was updated.
CREATOR	(varchar 10). Bank ID and Operator ID of the operator who created the record.
CREATE_DATE	(date). Date on which the record was created.
UPDATER	(varchar 10). Bank ID and Operator ID of the operator who updated the record.

UPDATE_DATE	(date). Date on which the record was updated.
VERIFIER	(varchar 10). Bank ID and Operator ID of operator who verified the record.
VERIFY_DATE	(date). Date on which the record was verified.
DAT_OWNER_PROD_ID	(varchar 3). Product ID.
DAT_OWNER_BANK_ID (fk)	(varchar 3). Bank ID.
DAT_OWNER_LOC	(varchar 6). Location ID.
DAT_OWNER_CUST	(varchar 20). Customer ID.
PROD_ID	(varchar 3). Product ID.
BANK_ID (fk)	(varchar 3). Bank ID.
LOC	(varchar 6). Location ID.
CUST	(varchar 20). Customer ID.
CFG_MEMO	(varchar 80). Line description of the configuration.
RECORD_EXPIRED	(varchar 16). Date and time that this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Configuration Item Table (CONFIG_ITEM_T)

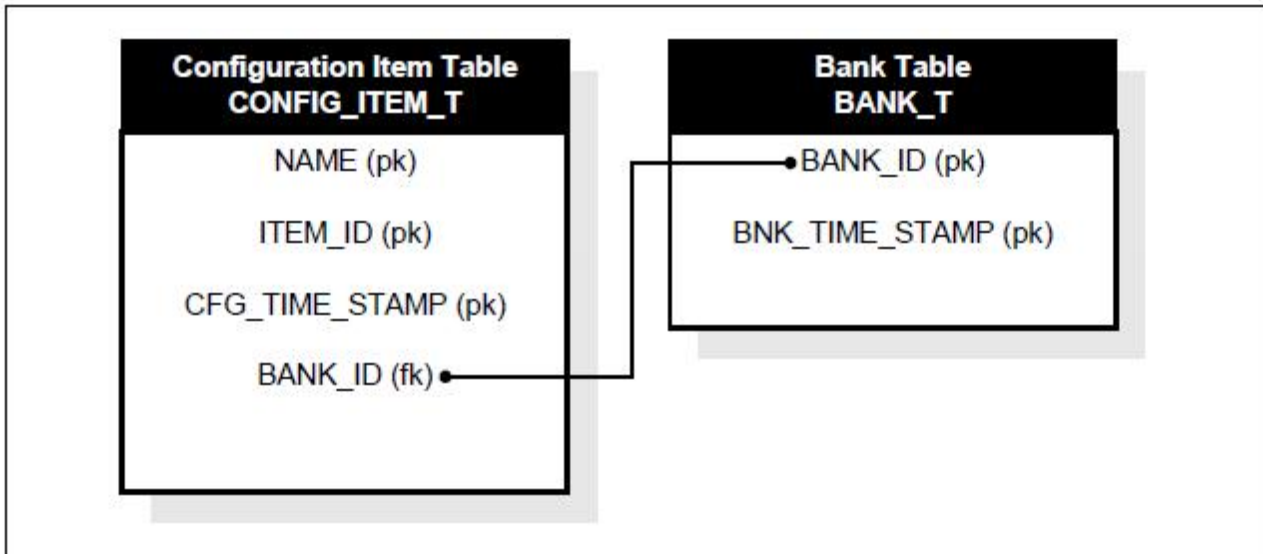
The Configuration Item table shows the individual table names for each table category listed in the Configuration table.

Table Relationships

The BANK_ID column in the Configuration Item table relates to the BANK_ID column in the Bank table.

Relationship Diagram

This relationship diagram shows the Configuration Item table along with the bank table.



Column Descriptions

NAME (pk)	(varchar 33). Name of the table or other configuration data being described.
ITEM_ID (pk)	(varchar 25). Configuration item being described.
CFG_TIME_STAMP (pk)	(varchar 16). Time record was updated.
PROD_ID	(varchar 3). Product ID.
BANK_ID (fk)	(varchar 3). Bank ID.

LOC	(varchar 6). Location ID.
CUST	(varchar 20). Customer ID.
DATATYPE	(varchar 16). Internal format of ASCII data value.
RECORD_EXPIRED	(varchar 16). Date and time at which the record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Configuration Item Description Table (CONFIG_ITEM_DESC_T)

Relating to the export of the configuration index, the Configuration Item Description table shows the association between a country code and a language. For a given configuration value, there is an associated country, and for that country, there is an associated language.

Primary Keys

The Configuration Item Description table (CONFIG_ITEM_DESC_T) has the following primary keys:

- NAME
- ITEM_ID
- TABLE_ORDINAL
- LANG_COUNTRY
- CFG_TIME_STAMP

Table Relationships

The CONFIG_ITEM_DESC_T contains no keys that relate to keys in other tables.

Column Descriptions

NAME (pk)	(varchar 33). Configuration ID.
ITEM_ID (pk)	(varchar 25). Configuration item ID.
TABLE_ORDINAL (pk)	(number 5). Sequence number of data values for this item.
CFG_TIME_STAMP (pk)	(varchar 16). Date and time record was updated.
PROD_ID	(varchar 3). Product ID.
BANK_ID (fk)	(varchar 3). Bank ID.
LOC	(varchar 6). Location ID.
CUST	(varchar 20). Customer ID.
LANG_COUNTRY	(varchar 5). Configuration language/country identifier code.

DESCRIPTION	(varchar 235). Description of the language/country identifier code.
RECORD_EXPIRED	(varchar 16). Date and time at which the record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Configuration Item Data Table (CONFIG_ITEM_DATA_T)

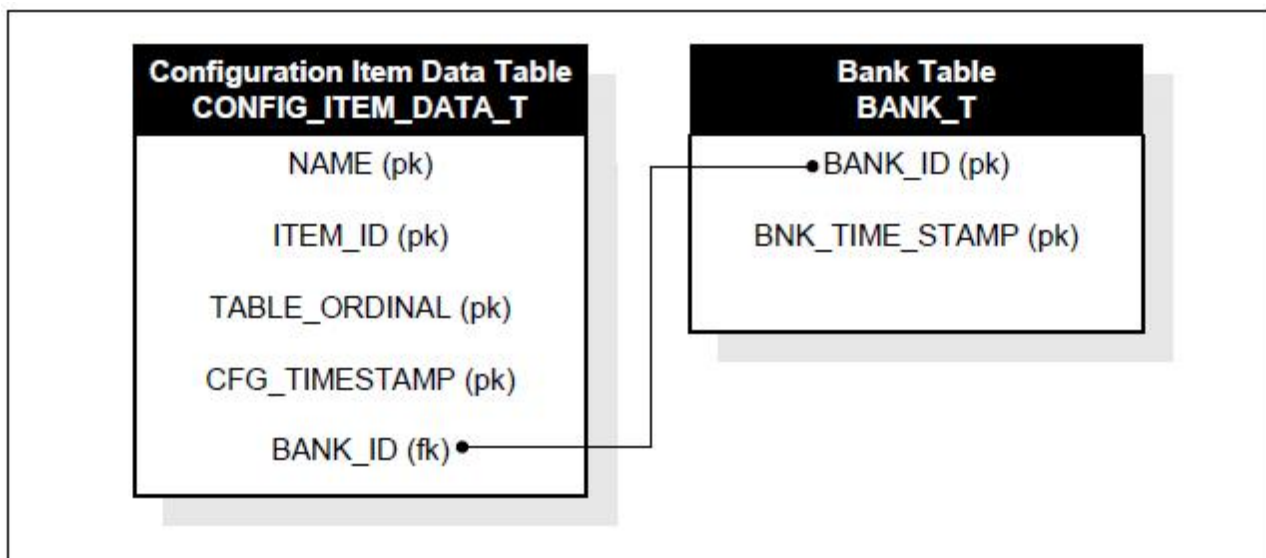
The Configuration Item Data table shows the actual table values for the tables listed in the Configuration Item table.

Table Relationships

The BANK_ID column in the Configuration Item Data table relates to the BANK_ID column in the Bank table.

Relationship Diagram

This relationship diagram shows the Configuration Item Data table along with the Bank table.



Column Descriptions

NAME (pk)	(varchar 33). Configuration ID.
ITEM_ID (pk)	(varchar 25). Configuration item ID.
TABLE_ORDINAL (pk)	(number 5). Sequence number of record.

CFG_TIME_STAMP (pk)	(varchar 16). Date and time record was updated.
PROD_ID	(varchar 3). Product ID.
BANK_ID (fk)	(varchar 3). Bank ID.
LOC	(varchar 6). Location ID.
CUST	(varchar 20). Customer ID.
TABLE_VALUE	(varchar 256). Data item (flag or table entry) that defines how the system is designed to operate.
RECORD_EXPIRED	(varchar 16). Date and time at which the record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Currency Table (CURRENCY_T)

The Currency table contains information about currency precisions.

Primary Keys

The Currency table (CURRENCY_T) has the following primary keys:

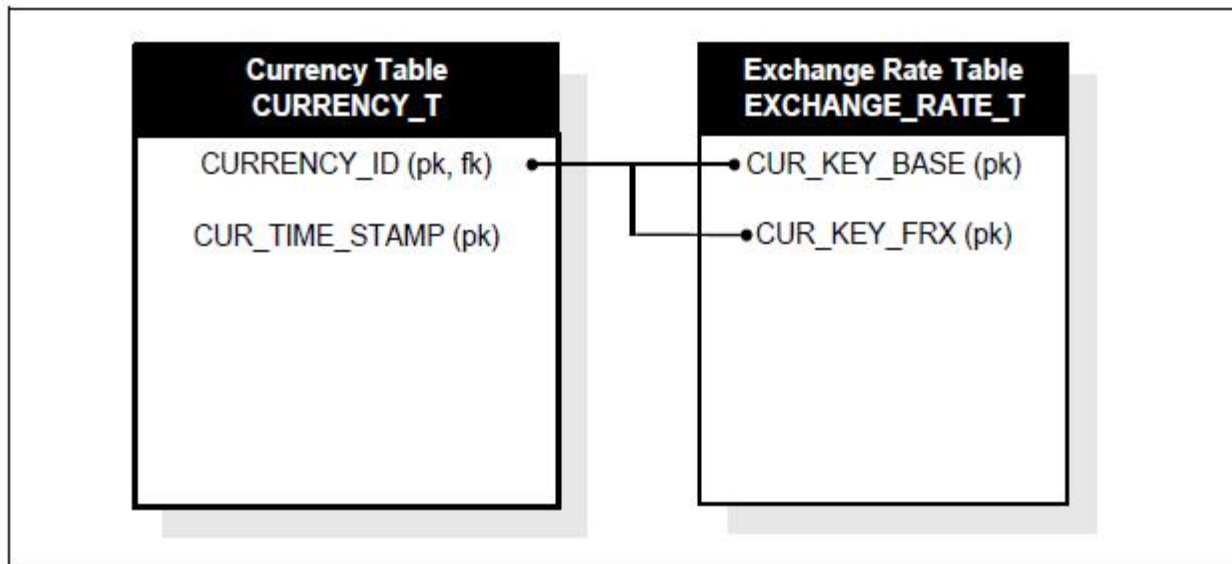
- CURRENCY_ID
- CUR_TIME_STAMP

Table Relationships

The CURRENCY_ID column in the Currency table relates to the CUR_KEY_BASE and CUR_KEY_FRX columns in the Exchange Rate table.

Relationship Diagram

This relationship diagram shows the Currency table and the Exchange Rate table.



Column Descriptions

CURRENCY_ID (pk, fk)	(varchar 3). ISO currency code.
CUR_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
CUR_DESC	(varchar 35). Identification of the currency
CUR_PRECISION	(varchar 1). Number of decimal places permitted.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

CUSIP Table (CUSIP_T)

The CUSIP table provides information about CUSIPs.

Primary Keys

The CUSIP table (CUSIP_T) has the following primary keys:

- CUSIIP_ID
- MATURITY_DATE
- CUSIP_DESCRIPTION

Due From Monitor Tables

Due From Monitor tables store extracted DFM data needed for activity reporting. The data needed for producing the reports was added to the RGW End Of Day extract process.

The Due-From Monitor function monitors balances in MTS accounts maintained at other banks. It automatically creates draw or cover transactions to ensure that those accounts maintain their target average balances. The Due-From Monitor function is useful when a bank agrees to keep a certain balance in an account at another bank. Four Due From Monitor tables store extracted DFM data needed for activity reporting. The data needed for producing the reports was added to the RGW End Of Day extract process. The four DFM tables are:

- Due-From Monitor table (DFM_T)
- Due-From Monitor Balance table (DFM_BAL_T)
- Due-From Monitor Derived table (DFM_DERIVED_T)
- Due-From Monitor TRN table (DFM_TRN_T)

Parent/child relationships

The DFM_T is the parent table to the following tables:

- DFM_BAL_T
- DFM_DERIVED_T
- DFM_TRN_T

Primary keys

All Due-From-Monitor tables have these primary keys in common:

- BANK_ID
- ID_TYPE
- ACCOUNT
- DFM_TIME_STAMP

Due From Monitor Table (DFM_T)

The Due From Monitor Table contains detailed data from the Due-from-Monitor set/registers.

Primary Keys

The Due From Monitor table (DFM_T) has the following primary keys:

- BANK_ID
- ID_TYPE
- DFM_TIME_STAMP
- ACCOUNT

Column Descriptions

BANK_ID (pk)	(varchar 3). Differentiator to allow configurable cross bank indices.
ID_TYPE (pk)	(varchar 1). Account ID type; contains the following values: <ul style="list-style-type: none">• D (DDA)• F (Foreign nostro account number)• G (General ledger account number)• P (CHIPS participant ID)• V (Savings account number)
ACCOUNT (pk)	(varchar 14). Account type and account ID.
DFM_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
MONITOR_COUNT	(number 5). Number of days, weeks, or months in the monitoring period.
MONITOR_PERIOD_TYPE	(char 1). Type of period during which the due-from account monitoring takes place: D - Daily, W - Weekly, M - Monthly. The DFM function monitors the average daily balance in a due-from account over the course of the monitoring period.
MONITOR_START_DAY	(char 2). Period type Day; the day on which the monitoring period begins.
MONITOR_START_MONTH	(char 2). Period type Month; the month in which the monitoring period begins.
DRAW_RPTV_ID	(varchar 8). Repetitive ID for automatic draws, or transfers that bring money in from an account at another bank.
COVER_RPTV_ID	(varchar 8). Repetitive ID for automatic covers, or transfers that send money out to the account at another bank.

CURRENT_TARGET	(number 21.3). Current target balance. This is the amount the bank agrees to keep as an average daily balance in the due-from account. In most cases, the balance is negative, indicating that money is kept in an account at another bank.
CURRENT_TARGET_DATE	(date). Effective date of the current target balance.
FUTURE_TARGET	(number 21.3). Future target balance. The future target value supersedes the current target on the future target effective date.
FUTURE_TARGET_DATE	(date). Effective date of the future target balance.
YTD_OVER_SHORT	(number 21.3). Amount of the year-to-date overage or shortage. This is the accumulated net divergence from targets so far during the current year, up to the beginning of the current monitoring period.
LAST_TRN	(varchar 12). Transaction reference number of the last draw or cover message.
LAST_RPTV	(varchar 8). Repetitive ID of the last draw or cover.
LAST_AMOUNT	(number 21.3). Amount of the last draw or cover.
LAST_TIME	(date). Timestamp for the last draw or cover.
EOD_CHANGE_DATE	(date). Date of the last manual change to end-of-day balances.
EOD_CHANGE_COUNT	(number 5). Serial number of the last manual change to end-of-day balances.
EOD_CHANGE_AMOUNT	(number 21.3). Amount of the last manual change to end-of-day balances.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(char 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Due From Monitor Balance Table (DFM_BAL_T)

The Due From Monitor Balance table (DFM_BAL_T) contains the balance history of the monitored account.

Primary Keys

The Due From Monitor Balance table (DFM_BAL_T) has the following primary keys:

- BANK_ID
- ID_TYPE
- ACCOUNT
- EOD_DATE
- DFM_TIME_STAMP

Column Descriptions

BANK_ID (pk)	(varchar 3). Differentiator to allow configurable cross bank indices.
ID_TYPE (pk)	(varchar 1). Account ID type; contains the following values: <ul style="list-style-type: none">• D (DDA)• F (Foreign nostro account number)• G (General ledger account number)• P (CHIPS participant ID)• V (Savings account number)
ACCOUNT (pk)	(varchar 14). Account type and ID.
EOD_DATE (pk)	(date). Date of this balance.
DFM_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
EOD_BALANCE	(number 21.3). Today's end-of-day balance.
OVER_SHORT	(number 21.3). Amount of today's overage or shortage. This is the difference between today's current net balance and the currently applied target amount for the account.

YTD_OVER_SHORT	(number 21.3). Amount of the year-to-date overage or shortage. This is the accumulated net divergence from targets so far during the current year, up to the beginning of the current monitoring period.
CHANGE_PERSON	(varchar 10). Change person.
CHANGE_COUNT	(number 5). Identifier that assures accuracy between the end-of-day change queues.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(char 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Due From Monitor Derived Table (DFM_DERIVED_T)

The Due From Monitor Derived table contains data calculated using the DFM set-up and balance logs. It provides detail and summary data.

Primary Keys

The Due From Monitor Derived table (DFM_DERIVED_T) has the following primary keys:

- BANK_ID
- ID_TYPE
- ACCOUNT
- DFM_TIME_STAMP

Column Descriptions

BANK_ID (pk)	(varchar 3). Differentiator to allow configurable cross bank indices.
ID_TYPE (pk)	(varchar 1). Account ID type; contains the following values: <ul style="list-style-type: none">• D (DDA)• F (Foreign nostro account number)• G (General ledger account number)• P (CHIPS participant ID)• V (Savings account number)
ACCOUNT (pk)	(varchar 14). Account type and ID.
DFM_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
PERIOD_START	(date). Date on which the reporting period starts.
PERIOD_END	(date). Date on which the reporting period ends.
ELAPSED_MON_DAYS	(number 5). Elapsed monitoring days; the number of days since the start of the monitoring period (not including today).
TOTAL_MON_DAYS	(number 5). Total number of days during which the due-from account has been monitored.

REMAINING_MON_DAYS	(number 5). Remaining monitoring days; the number of days remaining in the monitoring period (including today).
OPENING_BALANCE	(Number 21.3). Opening balance; the start-of-day balance for the account.
DEBIT_COUNT	(number 5). Count of today's debits so far.
CREDIT_COUNT	(number 5). Count of today's credits so far.
DEBIT_SUM	(number 21.3). The sum of today's debits so far.
CREDIT_SUM	(number 21.3). The sum of today's credits so far.
EOD_BALANCE	(number 21.3). Today's end-of-day balance.
OVER_SHORT	(number 21.3). Amount of today's overage or shortage. This is the difference between today's current net balance and the currently applied target amount for the account.
PTD_OVER_SHORT	(number 21.3). Accumulated net divergence from targets so far in the current monitoring period.
PROJECTION	(number 21.3). Account's daily average requirement for the rest of the monitoring period to meet the target currently applied. This figure and today's net balance are used to generate a suggested transfer amount.
ACCUM_OVER_SHORT	(number 21.3). Accumulated net divergence from targets so far this year, up to the beginning of the current monitoring period.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(char 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Due From Monitor Transactions Table (DFM_TRN_T)

The Due From Monitor TRN table contains transactions associated with the monitored account. It provides a list of transaction reference numbers that can be used to look up messages in the MESSAGE tables.

Primary Keys

The Due From Monitor TRN table (DFM_TRN_T) has the following primary keys:

- BANK_ID
- ID_TYPE
- ACCOUNT
- TRN_DATE
- TRN_NUMBER
- DFM_TIME_STAMP

Table Relationships

The TRN_DATE and TRN_NUMBER columns in the DFM Transactions table relate to the TRN_DATE and TRN_NUMBER columns in the following message tables:

- MESSAGE_T
- MESSAGE_CR_T
- MESSAGE_DR_T
- MESSAGE_ACCTG_T
- MESSAGE_DEST_T
- MESSAGE_HIST_T
- MESSAGE_QUEUE_T
- MESSAGE_TXT_T
- MESSAGE_REL_AMT_T

Column Descriptions

BANK_ID (pk)	(varchar 3). Differentiator to allow configurable cross bank indices.
--------------	---

ID_TYPE (pk)	(varchar 1). Account ID type; contains the following values: <ul style="list-style-type: none">• D (DDA).• F (Foreign nostro account number)• G (General ledger account number)• P (CHIPS participant ID)• V (Savings account number)
ACCOUNT (pk)	(varchar 14). Account type and ID.
TRN_DATE (pk)	(date). Processing date portion of the transaction number indicating the date when the message was created; from the TRN (Transaction Reference Number). .
TRN_NUMBER (pk)	(varchar 8). Serial number portion of the transaction number. Unique serial number assigned to the message by REFSERV (a background process in MTS that assigns the second half of the next consecutive transaction number). . The first transaction message of each processing day is 00000001.
DFM_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(char 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Exchange Rate Table (EXCHANGE_RATE_T)

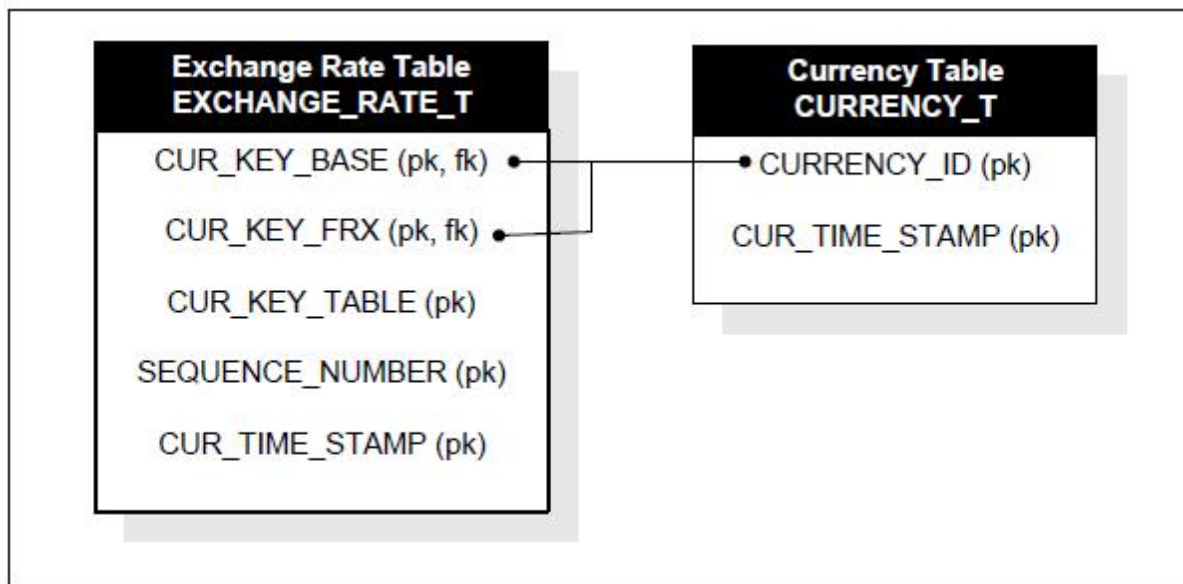
The Exchange Rate table provides information about currency exchange rates.

Table Relationship

The CUR_KEY_BASE and CUR_KEY_FRX columns in the Exchange Rate table relate to the CURRENCY_ID column in the Currency table.

Relationship Diagram

This relationship diagram shows the Exchange Rate table and the Currency table.



Column Descriptions

CUR_KEY_BASE (pk, fk)	(varchar 3). ISO currency code of the bank's base currency.
-----------------------	---

Exchange Rate Table (EXCHANGE_RATE_T)

CUR_KEY_FRX (pk, fk)	(varchar 3). ISO foreign currency code.
CUR_KEY_TABLE (pk)	(varchar 5). Exchange rate table identifier; identifies the table or tier used for a customer.
SEQUENCE_NUMBER (pk)	(number 10). A sequential number, generated by RGW, that uniquely identifies this record.
CUR_TIME_STAMP (pk)	(varchar 16). Date and time when the table entry for this currency was updated.
CUR_DESC	(varchar 35). Type and identification of the currency.
CUR_PRECISION	(varchar 1). Currency precision; number of decimal places permitted.
CUR_BUY_RT	(number 29.11). Rate at which a currency is bought.
CUR_SEL_RT	(number 29.11). Rate at which a currency is sold.
CUR_SPOT_RT	(number 29.11). The current exchange rate for this currency on the spot market. This column is not used for currency conversion umber.
CUR_30_DAY_RT	(number 29.11). Exchange rate for contracts involving this currency 30 days in the future. This column is not used for currency conversions.
CUR_BUY_LIMIT	(number 21.3). Amount in foreign currency above which MTS requires a trader control number in the Entry function's TRDR field.
CUR_SEL_LIMIT	(number 21.3). Amount in foreign currency above which MTS requires a trader control number in the Entry function's TRDR field.
CUR_TOLERANCE	(number 18.16). Determines the percentage by which the exchange rate in a transaction can deviate from the buy or sell rates in this table.
CUR_DT	(date). Date when the table entry for this currency was updated.
CUR_CODE (fk)	(varchar 4). Foreign exchange code; same as the ISO currency code. This column maps to the FX_CODE field on the CUR screen.
CUR_HOLD_FLAG	(varchar 1). Flag indicating whether MTS requires transactions involving this currency to be reviewed. This column contains the following values: Y (MTS holds debits or credits in this currency) or N (MTS does not hold debits or credits in this currency). This flag is used when a currency fluctuates regularly.
CUR_INDIRECT_FLAG	(varchar 1). Indicates how to display a foreign currency. This column contains the following values: I (indirect; MTS displays rates in the base currency) or D (direct; MTS displays rates in the foreign currency).
CUR_NO_EDIT	(varchar 1). Indicates whether you can change an exchange rate.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.

Exchange Rate Table (EXCHANGE_RATE_T)

RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
----------------	--

Holiday Table (HOLIDAY_T)

The Holiday table contains primary information about bank holiday tables that affect processing, including whether messages can be originated on a holiday, what clearing agents and districts are closed on the holiday, and a descriptive memo about the holiday.

Primary keys

The Holiday table (HOLIDAY_T) has the following primary keys:

- HOL_TABLE
- HOL_DATE
- HOL_COUNTRY
- SEQUENCE_NUM
- HOL_TIME_STAMP

Column Descriptions

HOL_TABLE (pk)	(varchar 25). Name of the holiday table.
HOL_DATE (pk)	(date). Date of the holiday.
HOL_COUNTRY (pk)	(varchar 2). ISO country code of the country for which this holiday applies.
SEQUENCE_NUM (pk)	(number 10). A sequential number, generated by RGW, that uniquely identifies this holiday.
HOL_TIME_STAMP (pk)	(varchar 16). Date and time when the holiday record was updated.
ORIG_RESTRICTED	(varchar 1). Flag indicating whether MTS allows messages to be originated from this country on this date (Y) or not (N).
CENTRAL_CLR_AGENT	(varchar 9). ABA routing number of a central clearing agent that is closed on this holiday.
MEMO	(varchar 80). Description of the holiday and any relevant comments.
RECORD_EXPIRED	(varchar 16). Date and time when this holiday record becomes invalid.

Holiday Table (HOLIDAY_T)

RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
PAYMENT_CHANNEL	(varchar 3). Holiday observed in this payment channel.

Monitor Table (MONITOR_T)

The Monitor table contains user-defined parameters that MTS matches against a payment. If there is a match, MTS performs the actions specified in the user-defined options (for example, route the payment to risk or print the message to a specified printer queue).

Table Relationships

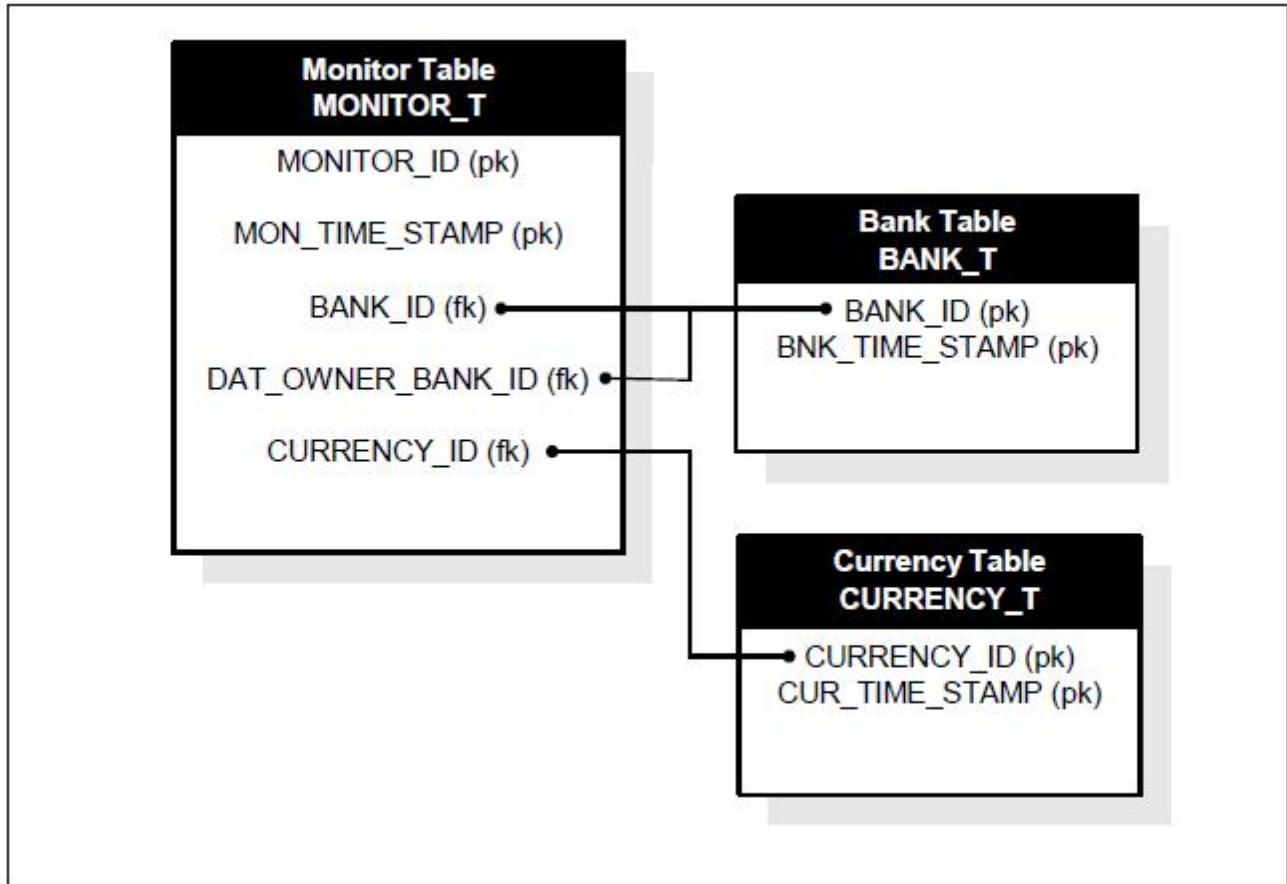
The following columns in the Monitor table relate to the BANK_ID column in the Bank table:

- BANK_ID
- DAT_OWNER_BANK_ID

The CURRENCY_ID column in the Monitor table relates to the CURRENCY_ID column in the Currency table.

Relationship Diagram

This relationship diagram shows the Monitor table along with the Bank and Currency tables, related to it through foreign keys.



Column Descriptions

MONITOR_ID (pk)	(varchar 20). Unique identifier associated with this Monitor file item; mapped from the MONITOR POINT ID field on the MON screen.
MON_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
DAT_FUNCTION	(varchar 7). Function that last accessed this account record; contains following values: VFY ADD (Verify add) VFY DEL (Verify delete) VFY UPD (Verify update)
CREATOR	(varchar 10). Operator ID and bank ID of the operator who created the monitor record.

CREATE_DATE	(date). Date when the monitor record was created.
UPDATER	(varchar 10). Operator ID and bank ID of the operator who updated the monitor record.
UPDATE_DATE	(date). Date when the monitor record was updated.
VERIFIER	(varchar 10). Operator ID and bank ID of the operator who verified the monitor record.
VERIFY_DATE	(date). Date when the monitor record was verified.
DAT_OWNER_PROD_ID	(varchar 3). Product ID.
DAT_OWNER_BANK_ID (fk)	(varchar 3). Bank ID.
DAT_OWNER_LOC	(varchar 6). Location ID.
DAT_OWNER_CUST	(varchar 20). Customer ID.
RISK_INTERCEPT	(varchar 1). Flag indicating whether MTS will intercept and route messages matching the monitoring criteria to the Risk queue for manual intervention (Y) or not (N).
BANK_ID (fk)	(varchar 3). Bank ID mapped from the BANK_ID field on the MON select screen.
LOW_AMOUNT	(number 21.3). Lower limit for the amount range to be monitored.
HIGH_AMOUNT	(number 21.3). Upper limit for the amount range to be monitored.
CURRENCY_ID (fk)	(varchar 3). Currency code for limit amounts.
START_TIM_HRS	(varchar 2). Hour portion of the time for the system to start monitoring.
START_TIM_MNS	(varchar 2). Minute portion of the time for the system to start monitoring.
END_TIM_HRS	(varchar 2). Hour portion of the time for the system to stop monitoring.
END_TIM_MNS	(varchar 2). Minute portion of the time for the system to stop monitoring.
TRAN_TYPE	(varchar 3). Transaction type; identifies the transaction type subject to monitoring. This column contains the following values: ANT (Anticipation) DEP (Deposit) DFA (Draft advice) DFT (Draft) DRR (Drawdown request) DRW (Drawdown response) FFB (Fed Funds bought) FFR (Fed Funds returned) FFS (Fed Funds sold)

Monitor Table (MONITOR_T)

	<p>FTR (Funds transfer)</p> <p>NON (Nonaccounting)</p> <p>PRE (Preadvised)</p> <p>SPL (Split accounting)</p>
SRC_CODE	<p>(varchar 3). Transaction source code. This column contains these values:</p> <p>ADJ (Adjustment)</p> <p>ADM (Administrative message entry function)</p> <p>CHP (CHIPS)</p> <p>CMS (Cash management system)</p> <p>DFM (Due from monitor)</p> <p>DOC (Letter of credit)</p> <p>ENT (Payment entry function)</p> <p>ETW (Enhanced Treasury Workstation). FED (Fedwire)</p> <p>INT (Internal transfer)</p> <p>LTC (Letter requiring callback)</p> <p>LTR (Letter)</p> <p>MEM (Memo post function)</p> <p>MIS (Miscellaneous, for example: cash management system or remote batch entry)</p> <p>MTW (Money Transfer Workstation)</p> <p>OFL (Offline)</p> <p>PHN (Phone transfer initiated outside bank)</p> <p>RBE (Remote batch entry)</p> <p>SOD (Start-of-day balance load)</p> <p>STO (Standing order function)</p> <p>SWF (S.W.I.F.T.)</p> <p>WIR (Telex)</p> <p>Any user-defined source</p>
DEST_CODE	<p>(varchar 3). Advice type of the message; contains the same values as for SRC_CODE above.</p>
MSG_TYPE	<p>(varchar 4). Message type associated with the transaction type.</p>
ABA_DISTRICT	<p>(varchar 9). For incoming messages with source FED, this column contains the Federal Reserve bank, district, or subdistrict that sent the message. For outgoing messages with source FED, this column contains the Federal Reserve bank, district, or subdistrict to which the message is sent. This column contains the following values:</p> <p>9-digit ABA number</p> <p>3-digit district code</p> <p>5-digit subdistrict code</p>
CREDIT_COUNTRY	<p>(varchar 2). ISO country code for the destination country.</p>

CREDIT_AREA_CODE	(varchar 4). Credit party area code.
DEBIT_COUNTRY	(varchar 2). ISO country code for the source country.
DEBIT_AREA_CODE	(varchar 4). Debit party area code.
MATCH_STR1	(varchar 35). First free-format data string to be checked against the message's debit and credit party fields. If a match occurs, MTS routes the message to Risk for manual intervention.
MATCH_STR2	(varchar 35). Second free-format data string to be checked against the message's debit and credit party fields. If a match occurs, MTS routes the message to Risk for manual intervention.
MATCH_STR3	(varchar 35). Third free-format data string to be checked against the message's debit and credit party fields. If a match occurs, MTS routes the message to Risk for manual intervention.
MATCH_STR4	(varchar 35). Fourth free-format data string to be checked against the message's debit and credit party fields. If a match occurs, MTS routes the message to Risk for manual intervention.
REQUESTOR	(varchar 35). Name of the person or department that requested the monitoring information. You can use spaces in this column, but no punctuation. This data does not appear on printed notifications.
ATTN	(varchar 35). Comment or the name of the person to be notified when a message meets monitoring criteria. This data appears on printed notifications. If the Risk intercept flag is set to Y, this data appears in the message history of intercepted transactions and in informational message system displays when you receive such a message from the Risk queue.
MEMO	(varchar 60). Internal memo text. This data does not appear on printed notifications.
QUEUE_PROD_ID	(varchar 3). Product ID part of a three-part queue name. Mapped from the PRODCUT ID field on the MON screen.
QUEUE_BANK_ID	(varchar 3). Bank part of a three-part queue name; Bank ID. Mapped from the BANK ID field on the MON screen.
QUEUE_LOC	(varchar 6). Location part of a three-part queue name; location of the designated printer.
QUEUE_CUST	(varchar 20). Customer ID part of a three-part queue name. Mapped from the CUSTOMER ID field on the MON screen.
QUEUE_NAME	(varchar 33). Queue name part of a three-part queue name; printer or processing queue to which notifications are to be routed.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UDPATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Monitor Table (MONITOR_T)

MATCH_PARTY	(varchar 1). Indicates whether to match debit, credit, or both party amounts.
CURRENCY_IND	(varchar 4). Currency indicator, either BASE or CURR.
STOP_INTERCEPT_FLAG	(varchar 1). If set to Y, and the monitor is hit, send to STOP review instead of RSK.

Operator Tables

The Operator tables contain information that enables supervisors to control the system privileges assigned to each MTS operator.

The Operator tables also contain information about adding and removing system operators and operator passwords and information about the functions that operators have performed during the day.

Parent/child relationships

The Operator table is the parent table to the following child table:

- OPERATOR_PRIV_T

Primary keys

All Operator tables have the following primary keys in common:

- OPERATOR_ID
- OPR_TIME_STAMP

Operator Table (OPERATOR_T)

The Operator table provides information about operator privileges for each operator who uses the MTS system.

Primary Keys

The Operator table (OPERATOR_T) has the following primary keys:

- OPERATOR_ID
- OPR_TIME_STAMP

Column Descriptions

OPERATOR_ID (pk)	(varchar 10). Operator ID.
OPR_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
DAT_FUNCTION	(varchar 7). Function that last accessed this operator record; contains the following values: VFY ADD (Verify add) VFY DEL (Verify delete) VFY UPD (Verify update)
CREATOR	(varchar 10). Operator ID and bank ID of the operator who created the record.
CREATE_DATE	(date). Date that the record was created.
UPDATER	(varchar 10). Operator ID and bank ID of the operator who updated the record.
UPDATE_DATE	(date). Date on which the item was updated.
VERIFIER	(varchar 10). Operator ID and bank ID of the operator who verified the record.
VERIFY_DATE	(date). Date that the record was verified.
DAT_OWNER_PROD_ID	(varchar 3). Product ID.
DAT_OWNER_BANK_ID	(varchar 3). Bank ID.
DAT_OWNER_LOC	(varchar 6). Location ID.

DAT_OWNER_CUST	(varchar 20). Customer ID.
LAST_LOGIN	(date). Date of the last login.
LAST_TRY	(date). Time of last failed login attempt.
LOGIN_ATTEMPTS	(number 5). Number of login attempts since the last successful login.
ENABLED_FLAG	(varchar 1). Indicates whether an operator's login has been enabled.
NAME	(varchar 50). Name of the operator.
PASSWORD_DATE	(date). Date the operator password expires; calculated from the PSWRD_GOOD_DAYS column in the Bank table.
ACCESS_MONDAY	(varchar 24). Reserved for custom use.
ACCESS_TUESDAY	(varchar 24). Reserved for custom use.
ACCESS_WEDNESDAY	(varchar 24). Reserved for custom use.
ACCESS_THURSDAY	(varchar 24). Reserved for custom use.
ACCESS_FRIDAY	(varchar 24). Reserved for custom use.
ACCESS_SATURDAY	(varchar 24). Reserved for custom use.
ACCESS_SUNDAY	(varchar 24). Reserved for custom use.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
OPR_EFFECTIVE_TIME	(date). Effective date and time of the operator.
OPR_UNION_TYPE	(varchar 8). Operator or template.
OPR_LEVEL	(char 1). Experience level of the operator, scaled from 0 through 9 with: 0 = Lowest experience, 9 = highest experience
OPR_SECURID_USERNAME	(varchar 32). Used in authorization requests to the RSA securID server

Operator Change Table (OPERATOR_CHANGE_T)

The Operator Change table contains operator activity information.

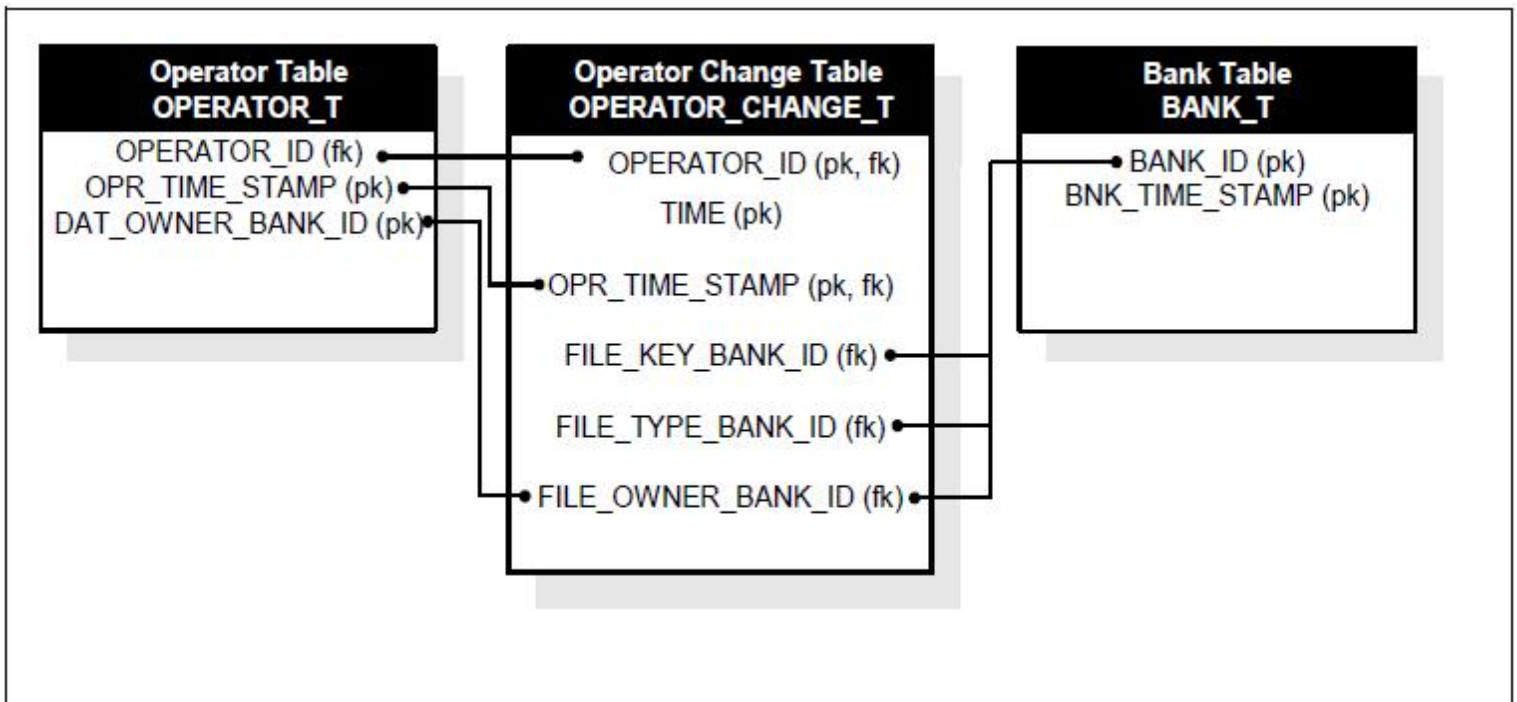
Table Relationships

The OPERATOR_ID column in the Operator table relates to the OPERATOR_ID column in the Operator Change table.

The FILE_KEY_BANK_ID, FILE_TYPE_BANK_ID, and FILE_OWNER_BANK_ID columns in the Operator table relate to the BANK_ID field in the Bank table.

Relationship Diagram

This relationship diagram shows the Operator Change table along with the Bank table.



Column Descriptions

OPERATOR_ID (pk, fk)	(varchar 12). ID of the operator who is using MTS.
----------------------	--

TIME (pk)	(date). System time when the action is performed.
OPR_TIME_STAMP (pk, fk)	(varchar 16). Date and time when the operator last accessed MTS.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
PERSON	(varchar 10). ID of the operator who performed a function.
AMOUNT	(number 21.3). Amount involved in the transaction.
TEXT	(varchar 20). Transaction text.
MEMO	(varchar 80). Descriptive text describing the operator action.
FILE_KEY_PROD_ID	(varchar 3). Product used by the operator who performed the action.
FILE_KEY_BANK_ID (fk)	(varchar 3). Bank ID of the bank where the operator uses MTS.
FILE_KEY_LOC	(varchar 6). Bank-defined location within the bank where the operator uses MTS.
FILE_KEY_CUST	(varchar 20). Customer ID.
FILE_KEY_NAME	(varchar 33). Customer name.
FILE_TYPE_PROD_ID	(varchar 3). Product used by the operator who performed the action.
FILE_TYPE_BANK_ID (fk)	(varchar 3). Bank ID of the operator who performed the action.
FILE_TYPE_LOC	(varchar 6). Location of the operator who performed the action.
FILE_TYPE_CUST	(varchar 20). Customer ID.
FILE_TYPE_NAME	(varchar 33). Name field from the primary index.
FILE_OWNER_PROD_ID	(varchar 3). Product used by the operator who performed the action.
FILE_OWNER_BANK_ID (fk)	(varchar 3). Bank ID of the operator who performed the action.
FILE_OWNER_LOC	(varchar 6). Location of the operator who performed the action.
FILE_OWNER_CUST	(varchar 20). Customer ID.
DAT_FUNC	(varchar 7). Function that last accessed this account record; contains following values: VFY ADD (Verify add) VFY DEL (Verify delete)

Operator Tables

	VFY UPD (Verify update)
--	-------------------------

Operator Error Table

(OPERATOR_ERROR_T)

The Operator Error table contains information regarding errors relating to specific operators.

Table Relationships

The OPERATOR_ID column in the Operator table relates to the OPERATOR_ID column in the Operator Error table.

The FILE_KEY_BANK_ID, FILE_TYPE_BANK_ID, and FILE_OWNER_BANK_ID columns in the Operator table relate to the BANK_ID field in the Bank table.

Column Descriptions

OPERATOR_ID (pk, fk)	(varchar 12). ID of the operator who is using MTS.
TIME (pk)	(date). System time when the action is performed.
OPR_TIME_STAMP (pk, fk)	(varchar 16). Date and time when the operator last accessed MTS.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
PERSON	(varchar 10). ID of the operator who performed a function.
AMOUNT	(number 21.3). Transaction amount.
TEXT	(varchar 20). Error text accompanying the operator action.
MEMO	(varchar 80). Descriptive text describing the operator action.
FILE_KEY_PROD_ID	(varchar 3). Product used by the operator who performed the action.
FILE_KEY_BANK_ID (fk)	(varchar 3). Bank ID of the bank where the operator performed the action.
FILE_KEY_LOC	(varchar 6). Bank-defined location within the bank where the operator uses MTS.
FILE_KEY_CUST	(varchar 20). Customer ID.
FILE_KEY_NAME	(varchar 33). Customer name.

Operator Tables

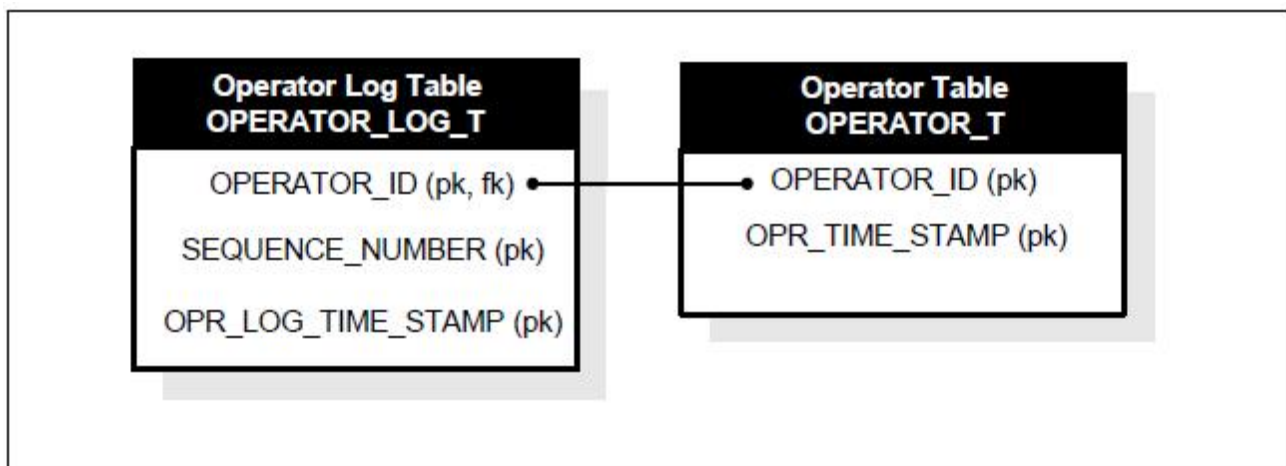
FILE_TYPE_PROD_ID	(varchar 3). Product used by the operator who performed the action.
FILE_TYPE_BANK_ID (fk)	(varchar 3). Bank ID of the operator who performed the action.
FILE_TYPE_LOC	(varchar 6). Location ID of the operator who performed the action.
FILE_TYPE_CUST	(varchar 20). Customer ID.
FILE_TYPE_NAME	(varchar 33). File name.
FILE_OWNER_PROD_ID	(varchar 3). Product used by the operator who performed the action.
FILE_OWNER_BANK_ID (fk)	(varchar 3). Bank ID of the operator who performed the action.
FILE_OWNER_LOC	(varchar 6). Location ID of the operator who performed the action.
FILE_OWNER_CUST	(varchar 20). Customer ID.
DAT_FUNC	(varchar 7). Function that the operator performed.

Operator Log Table (OPERATOR_LOG_T)

The Operator Log table contains information about functions performed by specific operators.

Relationship Diagram

This relationship diagram shows the Operator Log table along with its parent table, OPERATOR_T. The OPERATOR_ID field in the Operator Log table corresponds to the same field in the Operator table.



Column Descriptions

OPERATOR_ID (pk, fk)	(varchar 10). Operator who performed the action.
SEQUENCE_NUMBER (pk)	(number 10). Unique identifier of the transaction.
OPR_LOG_TIME_STAMP (pk)	(varchar 16). Date and time at which record was updated.
TIME	(date). Date the operator used the function.
PRODUCT	(varchar 3). Product used by the operator who performed the function.
FUNCTION	(varchar 6). Function used by the operator.
START_TIME	(date). Date when operator began performing the function.
END_TIME	(date). Date when operator stopped performing the function.

Operator Tables

OVERHEAD	(varchar 1). Contains Y for login or logout.
RECORD_EXPIRED	(varchar 16). Date and time at which this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Operator Logon Table (OPERATOR_LOGON_T)

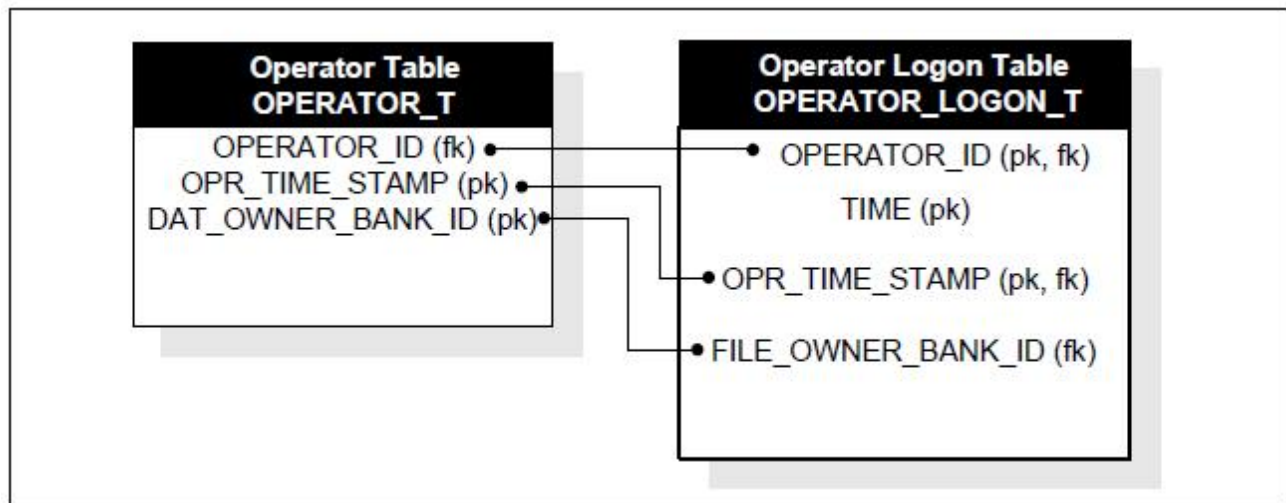
The Operator Logon table provides data for the Logon Failure report.

Table Relationships

The OPERATOR_ID and OPR_TIME_STAMP columns in the Operator Logon table relate to the same columns in the Operator table.

Relationship Diagram

This relationship diagram shows the Operator Logon table along with the Operator table, related to it through foreign keys.



Column Descriptions

OPERATOR_ID (pk, fk)	(varchar 12). Operator who performed the action.
TIME (pk)	(date). Date when the record was last updated.
OPR_TIME_STAMP (pk, fk)	(varchar 16). Date and time when the record was updated
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.

Operator Tables

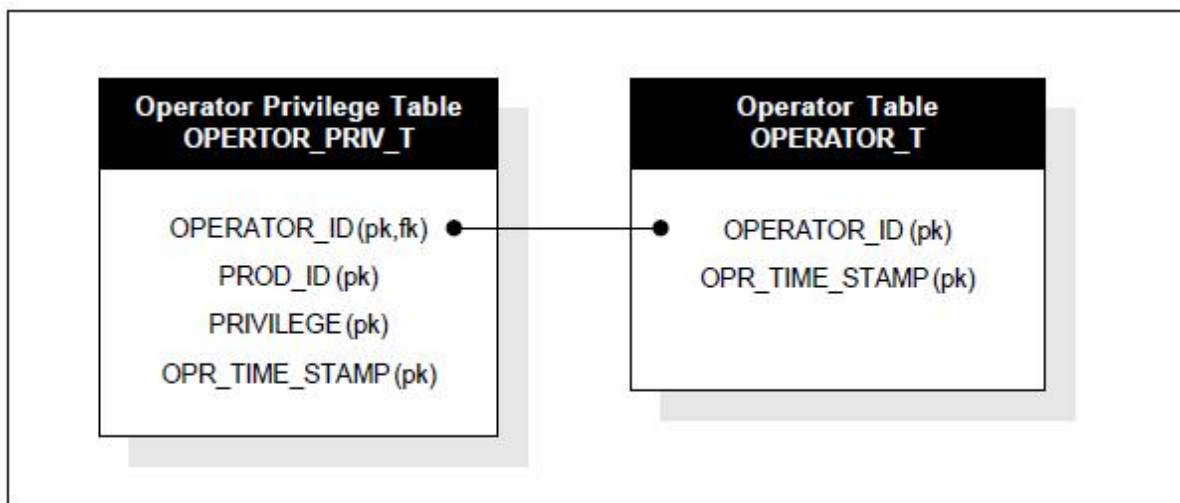
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
PERSON	(varchar 10). ID of the operator who performed a function.
AMOUNT	(number 21.3). Amount involved in a transaction.
TEXT	(varchar 20). Message text received or sent with a message.
MEMO	(varchar 80). Descriptive text describing the operator action.
FILE_KEY_PROD_ID	(varchar 3). Product used by the operator who performed the action.
FILE_KEY_BANK_ID	(varchar 3). Bank ID of the bank where the operator uses MTS.
FILE_KEY_LOC	(varchar 6). Bank-defined location within the bank where the operator uses MTS.
FILE_KEY_CUST	(varchar 20). Customer ID.
FILE_KEY_NAME	(varchar 33). Customer name.
FILE_TYPE_PROD_ID	(varchar 3). Product used by the operator who performed the action.
FILE_TYPE_BANK_ID (fk)	(varchar 3). Bank ID of the operator who performed the action.
FILE_TYPE_LOC	(varchar 6). Location of the operator who performed the action.
FILE_TYPE_CUST	(varchar 20). Customer ID.
FILE_TYPE_NAME	(varchar 33). Name field from the primary index.
FILE_OWNER_PROD_ID	(varchar 3). Product used by the operator who performed the action.
FILE_OWNER_BANK_ID (fk)	(varchar 3). Bank ID of the operator who performed the action.
FILE_OWNER_LOC	(varchar 6). Location of the operator who performed the action.
FILE_OWNER_CUST	(varchar 20). Customer ID.
DAT_FUNC	(varchar 7). Function that the operator performed.

Operator Privilege Table (OPERATOR_PRIV_T)

The Operator Privilege table provides information about all the functions and database privileges allowed an operator.

Relationship Diagram

This relationship diagram shows the Operator Privilege table along with its parent table, OPERATOR_T. The OPERATOR_ID field in the Operator Privilege field corresponds to the same field in the Operator table.



Column Descriptions

OPERATOR_ID (pk, fk)	(varchar 12). Operator ID.
PROD_ID (pk)	(varchar 3). Identifies the product that this operator can use.
PRIVILEGE (pk)	(varchar 8). List of privilege codes relating to MTS menu functions that the operator can use; mapped from the privilege codes listed in the FUNCTION or DATABASE screens in the OPR file.
OPR_TIME_STAMP (pk)	(varchar 16). Date and time when the record was updated.
SEQUENCE_NUM	(number 10). Record sequence number.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.

Operator Tables

RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
----------------	--

Operator Profile Amount Table (OPERATOR_PROFAMT_T)

The Operator Profile Amount table contains data about the high verify amount that an operator is granted, as it applies to their profile. An operator cannot verify a transfer if the amount exceeds the high verify amount that is defined on their profile.

Primary Keys

The Operator Profile Amount table (OPERATOR_PROFAMT_T) has the following primary keys:

- OPR_ID
- OPR_BANK
- OPR_LOC
- OPR_PROF_AMT_COD
- SEQUENCE_NUM
- OPR_TIME_STAMP

Column Descriptions

OPR_ID (pk)	(varchar 12). ID that identifies the operator.
OPR_BANK (pk)	(varchar 3). Operator's bank.
OPR_LOC (pk)	(varchar 6). Operator's location.
OPR_PROF_AMT_COD (pk)	(varchar 3). Operator's profile amount code.
SEQUENCE_NUM (pk)	(number 10). Unique identifier of this entry.
OPR_TIME_STAMP (pk)	(varchar 16). Time when the operator's record was updated.
OPR_PROF_AMT	(number 21.3). Operator's profile amount, or, the maximum amount of a transfer the operator can verify.
RECORD_EXPIRED	(varchar 16). Date and time when the profile record becomes invalid.

Operator Tables

RECORD_UPDATED (pk)	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
---------------------	--

Operator Template Table (OPERATOR_TMPL_T)

The Operator Template table identifies the location where the template for a particular operator is stored. It also identifies the template with which the operator ID is associated. A single operator may have multiple templates, one for each bank with whom the operator is associated.

Primary Keys

The Operator Template table (OPERATOR_TMPL_T). has the following primary keys:

- OPR_ID
- OPR_BANK
- OPR_LOC
- OPR_TPL_BANK
- OPR_TPL_LOC
- OPR_TIME_STAMP

Column Descriptions

OPR_ID (pk)	(varchar 12). Operator's ID.
OPR_BANK (pk)	(varchar 3). Operator's bank.
OPR_LOC (pk)	(varchar 6). Operator's location.
OPR_TPL_ID (pk)	(varchar 10). Operator's template ID.
OPR_TPL_BANK (pk)	(varchar 3). Operator's template bank.
OPR_TPL_LOC (pk)	(varchar 6). Operator's template location.
OPR_TIME_STAMP (pk)	(varchar 16). Time when the operator record was updated.
SEQUENCE_NUM	(number 10). Number (counter). that uniquely identifies this entry.
RECORD EXPIRED	(varchar 16). Date and time when the period becomes invalid.

Operator Tables

RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
----------------	--

Period Date Table (PERIOD_DATE_T)

The data in the Period Date table corresponds to the list of periods you see when you rerun RGW. A period is a processing time increment, usually set to a day.

Column Descriptions

PERIOD_DATE (pk)	(date) Date of the period when RGW was last run.
PERIOD_TIME_STAMP (pk)	(varchar 16) Date and time when the last RGW export was created.
RECORD_EXPIRED	(varchar 16) Date and time when the period becomes invalid.
RECORD_UPDATED	(varchar 1) Indicates whether this row's timestamp column has been propagated to previous rows.

PIN Tables

The PIN tables contain information associated with each caller authorized by a bank to perform phone-related money transfer operations. This information includes name, address, telephone information, test algorithm types, IDs of accounts for which the caller is authorized to transact business, transfer activities the caller is authorized to perform, and transaction limits for each function.

Parent/child relationships

The PIN table is the parent table to the following child tables:

- PIN_ACCOUNT_T
- PIN_FUNCTION_T
- PIN_FUNC_OVR_T

Primary keys

All PIN tables have the following primary keys in common:

- PIN_ID
- PIN_TIME_STAMP

PIN Table (PIN_T)

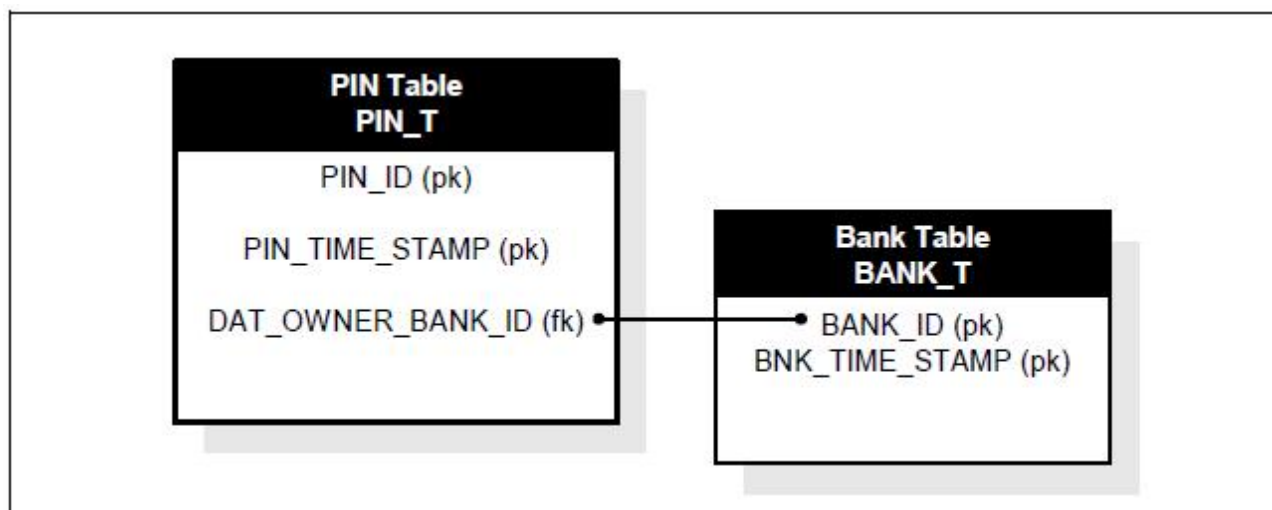
The PIN table contains caller name, address, telephone information, and test algorithm types (if used by the caller).

Bank table

The DAT_OWNER_BANK_ID field in the PIN table relates to the BANK_ID column in the Bank table.

Relationship Diagram

This relationship diagram shows the PIN table along with the tables related to it through foreign keys.



Column Descriptions

PIN_ID (pk)	(number 10). Sequential, system-generated reference identifier of this PIN record in a series of PIN records.
PIN_TIME_STAMP (pk)	(varchar 16). Date and time when this PIN record was created.
DAT_FUNCTION	(varchar 7). Function that last accessed this PIN record; contains the following values: VFY ADD (Verify add) VFY DEL (Verify delete)

PIN Tables

	VFY UPD (Verify update)
CREATOR	(varchar 10). Operator ID and bank ID of the operator who created the PIN record.
CREATE_DATE	(date). Date when the PIN record was created.
UPDATER	(varchar 10). Operator ID and bank ID of the operator who last updated the PIN record.
UPDATE_DATE	(date). Date when the PIN record was last updated.
VERIFIER	(varchar 10). Operator ID and bank ID of the operator who verified the record.
VERIFY_DATE	(date). Date when the PIN record was verified.
DAT_OWNER_PROD_ID	(varchar 3). Product ID.
DAT_OWNER_BANK_ID (fk).	(varchar 3). Bank ID.
DAT_OWNER_LOC	(varchar 6). Location ID.
DAT_OWNER_CUST	(varchar 20). Customer ID.
PIN_PREV_EFF_DATE	(date). Effective date for the previous PIN.
PIN_PREV_EXP_DATE	(date). Expiration date for the previous PIN.
PIN_CURR_EFF_DATE	(date). Effective date of the current PIN.
PIN_CURR_EXP_DATE	(date). Expiration date of the current PIN. A zero in this column means that the PIN never expires.
PIN_NEXT_EFF_DATE	(date). Effective date of the next PIN.
PIN_NEXT_EXP_DATE	(date). Expiration date of the next PIN. A zero in this column means that the PIN never expires.
PIN_LAST	(varchar 20). PIN owner's last name.
PIN_FIRST	(varchar 10). PIN owner's first name.
PIN_MIDDLE	(varchar 1). PIN owner's middle initial.
PIN_LINE1	(varchar 35). First line of PIN owner's address.
PIN_LINE2	(varchar 35). Second line of the PIN owner's address.
PIN_LINE3	(varchar 35). Third line of the PIN owner's address.

PIN_LINE4	(varchar 35). Fourth line of the PIN owner's address.
PIN_ZIP	(varchar 10). PIN owner's zip code.
PIN_COUNTRY	(varchar 2). PIN owner's country code.
PIN_PHONE	(varchar 20). PIN owner's phone number.
PIN_FAX	(varchar 20). PIN owner's fax number.
PIN_CUSTOMER_ID	(varchar 20). Customer ID that links this PIN owner with an account or address record in the REL file.
PIN_SPEC_INST1	(varchar 35). First line of the special instructions for a PIN.
PIN_SPEC_INST2	(varchar 35). Second line of the special instructions for a PIN.
PIN_SPEC_INST3	(varchar 35). Third line of the special instructions for a PIN.
PIN_SPEC_INST4	(varchar 35). Fourth line of the special instructions for a PIN.
PIN_MAIL_FLAG	(varchar 1). Not populated in this table.
PIN_MAIL_DATE	(date). Date that a secure mailer was mailed to the client, if appropriate.
RECORD_EXPIRED	(varchar 16). Date and time when the PIN record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
PIN_LAST_DATE	(date). Date when the PIN was last used.
PIN_LAST_REL_ID	(number 10). Last address record (REL ID) that was accessed using this PIN.
PIN_ACK_DATE	(date). Date when the future PIN (for future use) was acknowledged.
PIN_ACK_FLAG	(varchar 1). Flag indicating whether the newly mailed PIN has been acknowledged.
PIN_ACK_PERSON	(varchar 10). Operator ID and bank ID of the operator who acknowledged the PIN.
PIN_PIN_SECURID_ATTEMPTS	(number 5). Number of attempts to enter the SecurID.
PIN_DISABLED_FLAG	(char 1). Indicates the PIN disabled status.
PIN_SECURID_FLAG	(char 1). SecurID PIN and the token required for authorization.
PIN_SECURID_USERNAME	(varchar 32). Name used in the authorization request to the RSA SecurID server.

PIN Tables

PIN_USER_INFO	(varchar 35). Site specific data.
PIN_REL_REF_NUM	(number 10). The REL reference number for the name/address retrieval.
PIN_SECURID_EFF_DATE	(date). Effective date of the PIN secure ID token.
PIN_SECURID_EXP_DATE	(date). Expiration date of the PIN secure ID token.
PIN_SECURID_STATUS	(varchar 10). Status of the PIN secure ID token.
PIN_SECURID_DISABLED	(char 1). PIN secure ID token disabled.
PIN_SECURID_EFF1_DATE	(date). The next effective date of a PIN secure ID token.
PIN_SECURID_EXP1_DATE	(date). The next expiration date of a PIN secure ID token.

PIN Account Table (PIN_ACCOUNT_T)

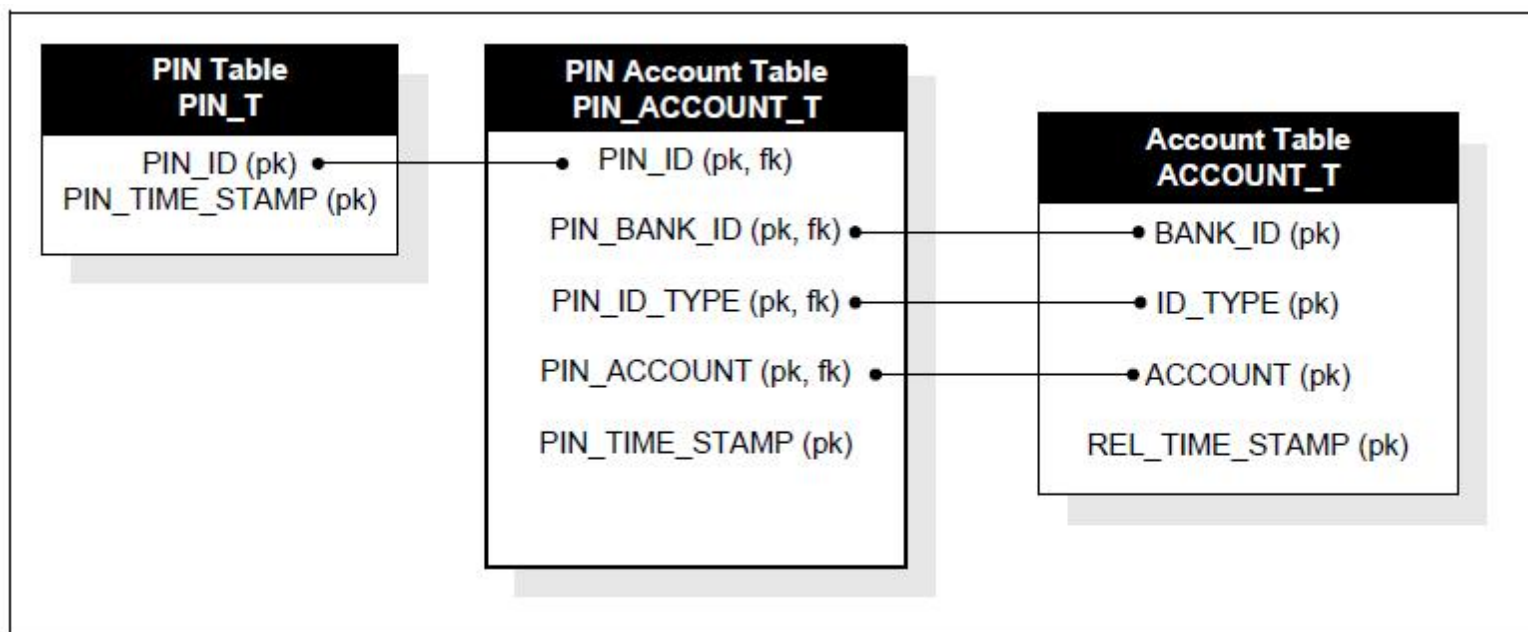
The PIN Account table provides information about the accounts for which an operator has privileges. Additionally, it contains account IDs and lists the transfer activities the caller is authorized to perform.

Table Relationships

The PIN_BANK_ID, PIN_ID_TYPE, and PIN_ACCOUNT columns in the PIN Account table relate to the BANK_ID, ID_TYPE, and ACCOUNT columns in the Account table.

Relationship Diagram

This relationship diagram shows the PIN Account table and the tables related to it through foreign keys.



Column Descriptions

PIN_ID (pk, fk)	(number 10). Sequential reference identifier for this PIN record in a series of PIN records.
PIN_BANK_ID (pk, fk)	(varchar 3). Bank identifier of the bank that owns the account tied to the PIN.

PIN Tables

PIN_ID_TYPE (pk, fk)	(varchar 1). PIN account ID type; contains the following values: B (Branch) D (DDA) F (Foreign nostro account number) G (General ledger account number) K (Customer) V (Savings account number) & (Department)
PIN_ACCOUNT (pk, fk)	(varchar 30). Account, branch, or department ID for which the PIN is valid.
PIN_TIME_STAMP (pk)	(varchar 16). Date and time when the PIN record was created.
PIN_SORT_CODE	(varchar 1). Priority for listing this caller on account-specific screens in the Phone Advising and Verify Callback functions. Numbers take priority over letters. This column contains the following values: Blank (Highest priority) 0-9 (Where 0 is the highest priority and 9 is the lowest) A-Z (Where A is the highest priority and Z is the lowest)
DEFAULT_PRIV_FLAG	(varchar 1). A flag indicating whether to override the bank's defaults for account privileges. An asterisk means to override bank defaults and apply account-specific functions and limits. A blank means to apply bank default functions and limits.
RECORD_EXPIRED	(varchar 16). Date and time when this PIN record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
PIN_ACC_CURRENCY	(varchar 3). Account currency.

PIN Function Override Table (PIN_FUNC_OVR_T)

The PIN Function Override table contains a list of default function overrides by account.

Primary Keys

The PIN Function Override table (PIN_FUNC_OVR_T) has the following primary keys:

- PIN_ID
- PIN_BANK_ID
- PIN_ID_TYPE
- PIN_ACCOUNT
- PIN_FUNCTION
- PIN_TIME_STAMP

Column Descriptions

PIN_ID (pk)	(number 10). Sequential, system-generated reference identifier of this PIN record in a series of PIN records.
PIN_BANK_ID (pk)	(varchar 3). Bank identifier of the bank that owns the account tied to the PIN.
PIN_ID_TYPE (pk)	(varchar 1). PIN account ID type; contains the following values: B (Branch) D (DDA) F (Foreign nostro account number) G (General ledger account number) K (Customer) V (Savings account number) & (Department)
PIN_ACCOUNT (pk)	(varchar 30). Account, branch, or department ID for which the PIN is valid.
PIN_FUNCTION (pk)	(varchar 6). Default function codes that determine which actions the PIN owner is allowed to perform across accounts. The values that appear here vary according to your configuration.

PIN Tables

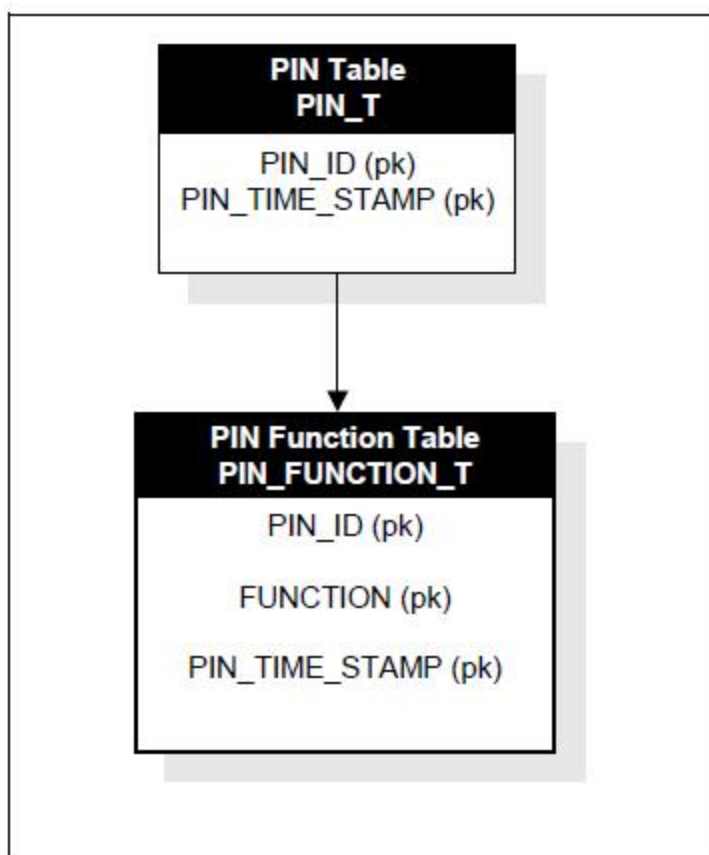
PIN_TIME_STAMP (pk)	(varchar 16). Date and time when the PIN record was created.
TRNSF_LIM	(number 21.3). Transaction limits for default functions. Transactions exceeding this limit are routed to a risk queue.
TRNSF_LIM_CURRENCY	(varchar 3). Transfer limit currency.
TEST_TYPE	(varchar 1). Test algorithm type; contains the following values: T (creates a test key from the algorithm specified in the Algorithm File) or 1-8 (creates a test key from an algorithm table in the bank-defined configuration file).
RECORD_EXPIRED	(varchar 16). Date and time when this PIN record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

PIN Function Table (PIN_FUNCTION_T)

The PIN Function table provides information about test algorithm types, transfer activities the caller is authorized to perform, and the transaction limits for each function.

Relationship Diagram

This relationship diagram shows the PIN Function table and its parent table, PIN_T.



Column Descriptions

PIN_ID (pK0	(number 10). Sequential reference identifier.
FUNCTION (pK0	(varchar 6). Default function codes that determine which actions the PIN owner is allowed to perform across accounts. The values that appear here vary according to your configuration.
PIN_TIME_STAMP (pK0	(varchar 16). Date and time when this record was created.

PIN Tables

TRNSF_LIM	(number 21.3). Transaction limits for default functions. Transactions exceeding this limit are routed to a risk queue.
TEST_TYPE	(varchar 1). Test algorithm type; contains the following values: T (creates a test key from the algorithm specified in the Algorithm File) or 1-8 (creates a test key from an algorithm table in the bank-defined configuration file).
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
TRNSF_LIM_CURRENCY	(varchar 3). Transfer limit currency.

Processing Rules Tables

Processing Rules Tables contain processing rules, or sets of instructions. Processing rules (formerly known as Standing Instructions) are associated with address, channel, or profile relationships, and are used to address a client's specific processing needs. They are defined in the Relationship File for MTS to use during payment processing.

Using the MTS processing rule feature, MTS can be defined to automatically override standard processing rules in a bank's configuration files. When MTS processes a payment for a client with processing rules on file, it applies these specific processing rules automatically.

Parent/child relationships

The Processing Rule table (PRULE_T) is the parent table to the Processing Rule Match table (PRULE_MATCH_T), the Processing Rule Parameters table (PRULE_PARAM_T), and to the Processing Rule Parameter Value table (PRULE_PARAM_VL_T).

Primary keys

The Processing Rules tables have the following primary keys in common:

- FUNC_AREA
- FUNC_AREA_ID
- BANK_ID
- PR_SEQ_NUM
- PR_TIME_STAMP

Processing Rule Table (PRULE_T)

The Processing Rule table contains a list of processing rules.

Processing rules are defined in the Address File for MTS to use during payment processing. If a client has unique processing rules on file, they are used in place of the standard processing method that would normally be used.

Primary Keys

The Processing Rule table (PRULE_T) has the following primary keys:

- FUNC_AREA
- FUNC_AREA_ID
- FUNC_AREA_ID_TYPE
- BANK_ID
- PR_SERIAL_ID
- PR_TIME_STAMP

Column Descriptions

FUNC_AREA (pk)	(varchar 3). Functional area of the database.
FUNC_AREA_ID (pk)	(varchar 30). ID within the functional area of the database.
FUNC_AREA_ID_TYPE (pk)	(char 1). ID type within the functional area of the database.
BANK_ID (pk)	(varchar 3). Bank ID. This makes the bank unique in a multibank environment.
PR_SERIAL_ID (pk)	(number 19). Processing rule sequence number.
PR_TIME_STAMP (pk)	(number 16). Date and time when the record was updated.
REL_ID	(number 10). Unique reference number that identifies this relationship.
PR_ORDINAL	(number 10). Processing rule ordinal.
PR_LEVEL	(varchar 20). Processing rule level.
PR_SOURCE	(varchar 20). Processing rule source.
PR_SOURCE_ID	(varchar 80). Identifies the source of the processing rule.
PR_NAME	(varchar 80). Name of the processing rule.
PR_TYPE	(varchar 80). Identifies the type of the processing rule.
PR_TEXT	(varchar 80). Text description of the processing rule.
PR_EFFECTIVE_TIME	(date). Date and time when the processing rule becomes effective. A value of 0 (zero). indicates it is effective immediately.
PR_EXPIRATION_TIME	(date). Date and time when the processing rule expires, or is no longer in effect. A value of 0 (zero). indicates the processing rule never expires.

PR_WHICH_SOURCE	(number 10). Number identifying the destination if the rule source is a destination.
PR_EVAL_STATUS	(varchar 80). Evaluation status of the processing rule.
PR_EVAL_WHEN	(date). Date on which the evaluation status last changed.
PR_EVAL_SERIAL	(number 10). Number indicating the serialization for the evaluation.
PR_EVAL_PREV_ORDINAL	(number 10). Number indicating the level at which the previous processing rule was evaluated.
PR_EVAL_PREV_COPY	(varchar 25). Reserved for future use.
PR_EVAL_PREV_LEVEL	(varchar 20). Previous level at which the processing rule was evaluated.
PR_EVAL_PREV_WHICH	(number 10). Destination number for the rule from the destination.
PR_EVAL_NEXT_ORDINAL	(number 10). Number indicating the level of the next rule to be evaluated.
PR_EVAL_NEXT_COPY	(varchar 25). Reserved for future use.
PR_EVAL_NEXT_LEVEL	(varchar 20). Name of the next processing rule level.
PR_EVAL_NEXT_WHICH	(number 10). Destination number for the rule from the destination.
PR_EVAL_MEMO	(varchar 80). Reserved for future use.
PR_SUBTYPE	(varchar 80). Rule subtype.
RECORD_EXPIRED	(number 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(char 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Processing Rules Match Table (PRULE_MATCH_T)

The Processing Rules Match table contains criteria used to match processing rules to a message. When there is a match, specific processing rules are invoked that control how the message is to be processed.

Primary Keys

The Processing Rules Match table (PRULE_MATCH_T) contains the following primary keys:

- FUNC_AREA
- FUNC_AREA_ID
- FUNC_AREA_ID_TYPE
- BANK_ID
- PR_SERIAL_ID
- PRMATCH_ID
- PRMATCH_COND
- PRMATCH_VALUE
- PR_TIME_STAMP

Column Descriptions

FUNC_AREA (pk)	(varchar 3). Functional area of the database.
FUNC_AREA_ID (pk)	(varchar 30). ID within the functional area of the database.
FUNC_AREA_ID_TYPE (pk)	(char 1). ID type within the functional area of the database.
BANK_ID (pk)	(varchar 3). Bank ID. This makes the bank unique in a multibank environment.
PR_SERIAL_ID (pk)	(number 10). Processing rule sequence number.
PRMATCH_ID (pk)	(varchar 40). Match ID.
PRMATCH_COND (pk)	(varchar 20). Match conditional.
PRMATCH_VALUE (pk)	(varchar 40). Match value.
PR_TIME_STAMP (pk)	(number 16). Date and time when the record was updated.
PRMATCH_LOGICAL	(varchar 10). Match logical.
PRMATCH_ACTION	(varchar 80). Describes the action to be taken when there is a match.
RECORD_EXPIRED	(number 16). Date and time when this record becomes invalid.

RECORD_UPDATED	(char 1). Indicates whether this row's timestamp column has been propagated to previous rows.
----------------	---

Processing Rules Parameter Table (PRULE_PARAM_T)

The Processing Rules Parameter table contains the parameters that are matched to messages to determine which processing rules will be used for processing.

Primary Keys

The Processing Rules Parameter table (PRULE_PARAM_T) has the following primary keys:

- FUNC_AREA
- FUNC_AREA_ID
- FUNC_AREA_ID_TYPE
- BANK_ID
- PR_SERIAL_ID
- PRPARAM_ID
- PR_TIME_STAMP

Column Descriptions

FUNC_AREA (pk)	(varchar 3). Functional area of the database.
FUNC_AREA_ID (pk)	(varchar 30). ID within the functional area of the database.
FUNC_AREA_ID_TYPE (pk)	(char 1). ID type within the functional area of the database.
BANK_ID (pk)	(varchar 3). Bank ID. This makes the bank unique in a multibank environment.
PR_SERIAL_ID (pk)	(number 19). Processing rule sequence number.
PRPARAM_ID (pk)	(varchar 40). Processing rules parameter ID.
PR_TIME_STAMP (pk)	(number 16). Date and time when the record was updated.
PRPARAM_EDIT	(varchar 20). Processing rule parameter edit type.
PRPARAM_COUNT	(number 10). Number of values in one row of the processing rule parameter table.
PRPARAM_SOURCE	(varchar 20). Description of the source of the processing rule.
PRPARAM_SOURCE_ID	(varchar 80). Identifies the source of the processing rule.
RECORD_EXPIRED	(number 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(char 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Processing Rules Parameter Values Table (PRULE_PARAM_VL_T)

The Processing Rules Parameter Values table contains the values that match the parameters used in determining which processing rules are used to process a message.

Primary Keys

The Processing Rules Parameter Values table (PRULE_PARAM_VL_T) table has the following primary keys:

- FUNC_AREA
- FUNC_AREA_ID
- FUNC_AREA_ID_TYPE
- BANK_ID
- PR_SERIAL_ID
- PARAM_ID
- VL_SERIAL_ID
- PR_TIME_STAMP

Column Descriptions

FUNC_AREA (pk)	(varchar 3). Functional area of the database.
FUNC_AREA_ID (pk)	(varchar 30). ID within the functional area of the database.
FUNC_AREA_ID_TYPE (pk)	(char 1). ID type within the functional area of the database.
BANK_ID (pk)	varchar 3). Bank ID. This makes the bank unique in a multibank environment.
PR_SERIAL_ID (pk)	(number 19). Processing rule sequence number.
PRPARAM_ID (pk)	(varchar 40). Processing rules parameter ID.
VL_SERIAL_ID (pk)	(number 19). Value sequence number.
PR_TIME_STAMP (pk)	(number 16). Date and time when the record was updated.
PARAMETER_VALUES	(varchar 80). Value of the parameters.
RECORD_EXPIRED	(number 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(char 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Profile Table (PROFILE_T)

The Profile table contains information that defines a category of clients. Similar to a template, profiles simplify maintaining records for multiple clients that are identified with a particular profile.

Primary Keys

The profile table (PROFILE_T) has the following primary keys:

- PROFILE_ID
- PROFILE_TIME_STAMP

Column Descriptions

PROFILE_ID (pk)	(varchar 24). Profile ID.
PROFILE_TIME_STAMP (pk)	(number 16). Date and time when the profile record was updated.
REL_ID	(number 10). Unique reference number that identifies this relationship.
BANK_ID (pk)	(varchar 3). Profile bank. This makes the bank unique in a multibank environment.
DAT_FUNCTION	(varchar 7). Function that last accessed this profile record; contains following values: VFY ADD (Verify add) VFY DEL (Verify delete) VFY UPD (Verify update)
CREATOR	(varchar 10). Operator ID and bank ID of the operator who created the record.
CREATE_DATE	(date). Date when the record was created.
UPDATER	(varchar 10). Operator ID and bank ID of the operator who last updated the record.
UPDATE_DATE	(date). Date on which the record was updated.

VERIFIER	(varchar 10). Operator ID and bank ID of the operator who verified the record.
VERIFY_DATE	(date). Date that the record was verified.
DAT_OWNER_PROD_ID	(varchar 3). Product ID.
DAT_OWNER_BANK_ID	(varchar 3). Bank ID.
DAT_OWNER_LOC	(varchar 6). Location ID.
DAT_OWNER_CUST	(varchar 20). Customer ID.
PROFILE_NAME	(varchar 35). Profile name.
PROFILE_DESC	(varchar 35). Description of the profile.
PROFILE_EFF_DATE	(date). Effective date of the profile.
PROFILE_EXP_DATE	(date). Expiration date of the profile.
DOCUMENT_ID	(varchar 6). Document ID.
RECORD_EXPIRED	(number 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

PSR Table (PSR_T)

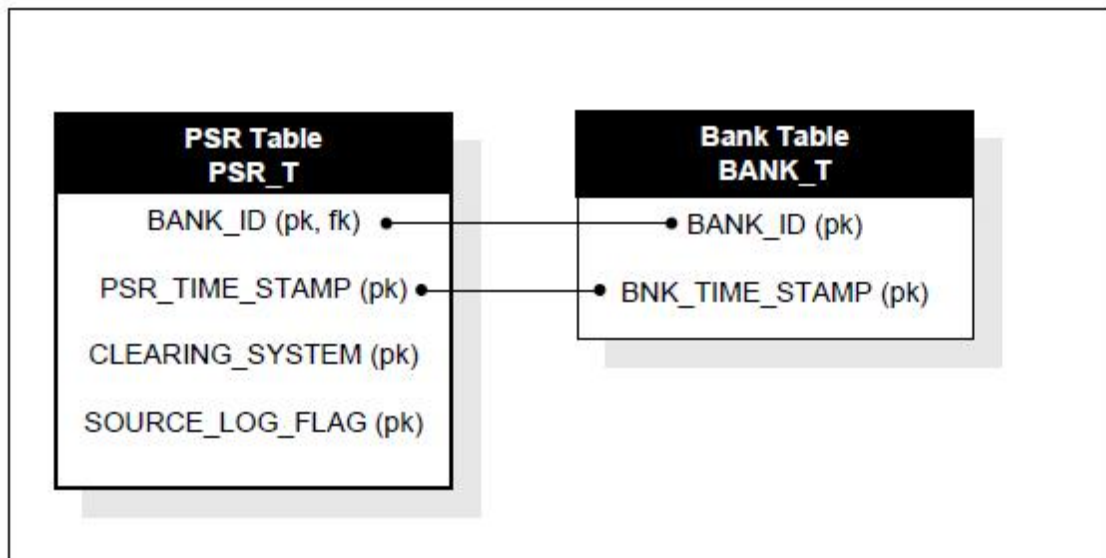
The PSR table contains monitor and control information that reflects a bank's payment network positions and that is used to evaluate a bank's overall position.

Table Relationships

The BANK_ID column in the PSR table relates to the BANK_ID column in the Bank table.

Relationship Diagram

This relationship diagram shows the PSR table along with the tables related to it through foreign keys.



Column Descriptions

BANK_ID (pk, fk)	(varchar 3). Bank identifier of the bank that owns this account.
CLEARING_SYSTEM (pk)	(varchar 20). The clearing system that is being monitored.

PSR_TIME_STAMP (pk)	(varchar 16). Date and time when the record was created.
SOURCE_LOG_FLAG (pk)	(char 1). Flag indicating the source of the update. Values are: C = Change log I = Intraday log
SAFETY_FACTOR	(number 18.3). Factor that allows a bank to save a percentage of its capital; mapped from PSR screen 3.
CAPITAL	(number 21.3). Amount of the adjusted bank capital; mapped from the CAPITAL field on PSR screen 8.
SINGLE_MULT	(number 18.3). Single day class factor, determined by the bank. The value in the Capital column is multiplied by this number to determine the single-day debit cap. This value is mapped from the X CLASS FACTOR/SINGLE DAY CAP field on PSR screen 8.
TWO_WEEK_MULT	(number 18.3). Two week class factor, determined by the bank. The value in the Capital column is multiplied by this number to determine the two-week debit cap. This value is mapped from the X CLASS FACTOR/TWO WEEK CAP field on PSR screen 8.
BUSINESS_DAYS	(number 18.3). Number of business days in the monitoring period; this is usually 10 days but can be less due to holidays.
OVERDRAFT_Q_AMT	(number 21.3). Amount that a bank is overdrawn.
FED_THROTTLE_IND	(varchar 13). Federal throttle setting and reason code. A throttle instructs MTS to hold outgoing payments within PSR to avoid exceeding Federal Reserve overdraft limits. This column contains the following values: OFF-NORMAL (Throttle is off because no limit has been exceeded; MTS will continue checking). ON-1 DAY (Throttle is on because the 1-day available funds limit has been exceeded). ON-2 WEEK (Throttle is on because the 2-week available funds limit has been exceeded). ON-INTERNAL (Throttle is on because the internal cap has been exceeded). ON-DEBIT CAP (Throttle is on because the debit cap has been exceeded). ON-CROSS SYS (Throttle is on because the cross system cap has been exceeded). ON-FORCED (Throttle is on because of an operator override command; checking is disabled). OFF-FORCED (Throttle is off because of an operator override command; checking is disabled). ON_CHP DEFER (Fed throttle is on because the CHIPS throttle is on). ON-EXTERNAL (Throttle is on because an external cap has been exceeded).
FED_DBT_CAP	(number 21.3). Bank's maximum overdraft allowance with the Federal Reserve for a specific time period.

PSR Table (PSR_T)

FED_ALARM_FACTOR	(number 21.3). Not populated in this table.
FED_INTERNAL_CAP	(number 21.3). Self-imposed debit cap with the Federal Reserve.
FED_THRESH	(number 21.3). Limit bank sets to hold individual transfers that exceed a specific amount.
CHP_THROTTLE_IND	(varchar 13). MTS holds outgoing payments above this amount within PSR to avoid exceeding the CHIPS overdraft limits.
CHP_DBT_CAP	(number 21.3). Bank's maximum overdraft allowance with CHIPS for a specific time period.
CHP_ALARM_FACTOR	(number 21.3). Threshold that instructs MTS to indicate when a certain percentage of the bank's available CHIPS funds have been used.
CHP_INTERNAL_CAP	(number 21.3). Self-imposed debit cap with CHIPS.
CHP_THRESH	(number 21.3). Threshold amount set by bank; MTS holds individual transfers that exceed this amount.
FED_OPEN_BAL	(number 21.3). Bank's opening balance with the Federal Reserve.
FEDWIRE_CDT_AMT	(number 21.3). Total amount of incoming Fedwire payments.
FEDWIRE_DBT_AMT	(number 21.3). Total amount of outgoing Fedwire payments.
SHARES_CDT_AMT	(number 21.3). Total amount of incoming book entry security items.
SHARES_DBT_AMT	(number 21.3). Total amount of outgoing book entry security items.
CHIPS_CDT_AMT	(number 21.3). Total amount of incoming CHIPS payments.
CHIPS_DBT_AMT	(number 21.3). Total amount of outgoing CHIPS payments.
OFFLINE_CDT	(number 21.3). Total credits of all offline categories.
OFFLINE_DBT	(number 21.3). Total debits of all offline categories.
OFFLINE_NET	(number 21.3). The cash net of all offline categories; mapped from PSR screen 7.
OFFLINE_NET2	(number 21.3). The locally calculated adjusted offline net.
OFFLINE_ABMS	(number 21.3). The adjusted offline net, as calculated by the Fed. This is the remotely calculated figure of the adjusted offline net.
OFFLINE_USED	(number 21.3). Either the locally calculated adjusted offline net or the remotely calculated adjusted offline net, depending upon how MTS is configured.

OFFL_01_CDT	(number 21.3). Offline category #1 credit amount. The OFFL_01_CDT through OFFL_30_CDT column and OFFL_01_DBT through OFFL_30_DBT columns are used by banks to store offline information. Depending on the bank's configuration, this information can be factored into available funds or not.
OFFL_01_DBT	(number 21.3). Offline category #1 debit amount.
OFFL_02_CDT	(number 21.3). Offline category #2 credit amount.
OFFL_02_DBT	(number 21.3). Offline category #2 debit amount.
OFFL_03_CDT	(number 21.3). Offline category #3 credit amount.
OFFL_03_DBT	(number 21.3). Offline category #3 debit amount.
OFFL_04_CDT	(number 21.3). Offline category #4 credit amount.
OFFL_04_DBT	(number 21.3). Offline category #4 debit amount.
OFFL_05_CDT	(number 21.3). Offline category #5 credit amount.
OFFL_05_DBT	(number 21.3). Offline category #5 debit amount.
OFFL_06_CDT	(number 21.3). Offline category #6 credit amount.
OFFL_06_DBT	(number 21.3). Offline category #6 debit amount.
OFFL_07_CDT	(number 21.3). Offline category #7 credit amount.
OFFL_07_DBT	(number 21.3). Offline category #7 debit amount.
OFFL_08_CDT	(number 21.3). Offline category #8 credit amount.
OFFL_08_DBT	(number 21.3). Offline category #8 debit amount
OFFL_09_CDT	(number 21.3). Offline category #9 credit amount.
OFFL_09_DBT	(number 21.3). Offline category #9 debit amount.
OFFL_10_CDT	(number 21.3). Offline category #10 credit amount.
OFFL_10_DBT	(number 21.3). Offline category #10 debit amount.
OFFL_11_CDT	(number 21.3). Offline category #11 credit amount.
OFFL_11_DBT	(number 21.3). Offline category #11 debit amount.
OFFL_12_CDT	(number 21.3). Offline category #12 credit amount.
OFFL_12_DBT	(number 21.3). Offline category #12 debit amount.

PSR Table (PSR_T)

OFFL_13_CDT	(number 21.3). Offline category #13 credit amount.
OFFL_13_DBT	(number 21.3). Offline category #13 debit amount.
OFFL_14_CDT	(number 21.3). Offline category #14 credit amount.
OFFL_14_DBT	(number 21.3). Offline category #14 debit amount.
OFFL_15_CDT	(number 21.3). Offline category #15 credit amount.
OFFL_15_DBT	(number 21.3). Offline category #15 debit amount.
OFFL_16_CDT	(number 21.3). Offline category #16 credit amount.
OFFL_16_DBT	(number 21.3). Offline category #16 debit amount.
OFFL_17_CDT	(number 21.3). Offline category #17 credit amount.
OFFL_17_DBT	(number 21.3). Offline category #17 debit amount.
OFFL_18_CDT	(number 21.3). Offline category #18 credit amount.
OFFL_18_DBT	(number 21.3). Offline category #18 debit amount.
OFFL_19_CDT	(number 21.3). Offline category #19 credit amount.
OFFL_19_DBT	(number 21.3). Offline category #19 debit amount.
OFFL_20_CDT	(number 21.3). Offline category #20 credit amount.
OFFL_20_DBT	(number 21.3). Offline category #20 debit amount.
OFFL_21_CDT	(number 21.3). Offline category #21 credit amount.
OFFL_21_DBT	(number 21.3). Offline category #21 debit amount.
OFFL_22_CDT	(number 21.3). Offline category #22 credit amount.
OFFL_22_DBT	(number 21.3). Offline category #22 debit amount.
OFFL_23_CDT	(number 21.3). Offline category #23 credit amount.
OFFL_23_DBT	(number 21.3). Offline category #23 debit amount.
OFFL_24_CDT	(number 21.3). Offline category #24 credit amount.
OFFL_24_DBT	(number 21.3). Offline category #24 debit amount.
OFFL_25_CDT	(number 21.3). Offline category #25 credit amount.

OFFL_25_DBT	(number 21.3). Offline category #25 debit amount.
OFFL_26_CDT	(number 21.3). Offline category #26 credit amount.
OFFL_26_DBT	(number 21.3). Offline category #26 debit amount.
OFFL_27_CDT	(number 21.3). Offline category #27 credit amount.
OFFL_27_DBT	(number 21.3). Offline category #27 debit amount.
OFFL_28_CDT	(number 21.3). Offline category #28 credit amount.
OFFL_28_DBT	(number 21.3). Offline category #28 debit amount.
OFFL_29_CDT	(number 21.3). Offline category #29 credit amount.
OFFL_29_DBT	(number 21.3). Offline category #29 debit amount.
OFFL_30_CDT	(number 21.3). Offline category #30 credit amount.
OFFL_30_DBT	(number 21.3). Offline category #30 debit amount.
BOOK_ENT_INT	(number 21.3). Current book entry security amount.
BOOK_ENT_MAT	(number 21.3). Current book entry maturity amount.
AUCTION_PURCH	(number 21.3). Current amount of auction purchases.
STABLE_COLLAT	(number 21.3). Current amount of stable collateral.
COLLAT_PLDG_ELSE	(number 21.3). Amount of collateral pledged elsewhere.
IN_TRANSIT_SEC	(number 21.3). Amount of securities in transit.
FED_RESERVE_TARGET	(number 21.3). Federal Reserve target amount.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
OFFL_UNVAL_ABMS	(number 21.3). Fed unavailable offline ABMS report balance.
OFFL_01_PRJ_CDT	(number 21.3). Offline category #01 projected credit amount.
OFFL_01_PRJ_DBT	(number 21.3). Offline category #01 projected debit amount.
OFFL_02_PRJ_CDT	(number 21.3). Offline category #02 projected credit amount.
OFFL_02_PRJ_DBT	(number 21.3). Offline category #02 projected debit amount.

PSR Table (PSR_T)

OFFL_03_PRJ_CDT	(number 21.3). Offline category #03 projected credit amount.
OFFL_03_PRJ_DBT	(number 21.3). Offline category #03 projected debit amount.
OFFL_04_PRJ_CDT	(number 21.3). Offline category #04 projected credit amount.
OFFL_04_PRJ_DBT	(number 21.3). Offline category #04 projected debit amount.
OFFL_05_PRJ_CDT	(number 21.3). Offline category #05 projected credit amount.
OFFL_05_PRJ_DBT	(number 21.3). Offline category #05 projected debit amount.
OFFL_06_PRJ_CDT	(number 21.3). Offline category #06 projected credit amount.
OFFL_06_PRJ_DBT	(number 21.3). Offline category #06 projected debit amount.
OFFL_07_PRJ_CDT	(number 21.3). Offline category #07 projected credit amount.
OFFL_07_PRJ_DBT	(number 21.3). Offline category #07 projected debit amount.
OFFL_08_PRJ_CDT	(number 21.3). Offline category #08 projected credit amount.
OFFL_08_PRJ_DBT	(number 21.3). Offline category #08 projected debit amount.
OFFL_09_PRJ_CDT	(number 21.3). Offline category #09 projected credit amount.
OFFL_09_PRJ_DBT	(number 21.3). Offline category #09 projected debit amount.
OFFL_10_PRJ_CDT	(number 21.3). Offline category #10 projected credit amount.
OFFL_10_PRJ_DBT	(number 21.3). Offline category #10 projected debit amount.
OFFL_11_PRJ_CDT	(number 21.3). Offline category #11 projected credit amount.
OFFL_11_PRJ_DBT	(number 21.3). Offline category #11 projected debit amount.
OFFL_12_PRJ_CDT	(number 21.3). Offline category #12 projected credit amount.
OFFL_12_PRJ_DBT	(number 21.3). Offline category #12 projected debit amount.
OFFL_13_PRJ_CDT	(number 21.3). Offline category #13 projected credit amount.
OFFL_13_PRJ_DBT	(number 21.3). Offline category #13 projected debit amount.
OFFL_14_PRJ_CDT	(number 21.3). Offline category #14 projected credit amount.
OFFL_14_PRJ_DBT	(number 21.3). Offline category #14 projected debit amount.
OFFL_15_PRJ_CDT	(number 21.3). Offline category #15 projected credit amount.

OFFL_15_PRJ_DBT	(number 21.3). Offline category #15 projected debit amount.
OFFL_16_PRJ_CDT	(number 21.3). Offline category #16 projected credit amount.
OFFL_16_PRJ_DBT	(number 21.3). Offline category #16 projected debit amount.
OFFL_17_PRJ_CDT	(number 21.3). Offline category #17 projected credit amount.
OFFL_17_PRJ_DBT	(number 21.3). Offline category #17 projected debit amount.
OFFL_18_PRJ_CDT	(number 21.3). Offline category #18 projected credit amount.
OFFL_18_PRJ_DBT	(number 21.3). Offline category #18 projected debit amount.
OFFL_19_PRJ_CDT	(number 21.3). Offline category #19 projected credit amount.
OFFL_19_PRJ_DBT	(number 21.3). Offline category #19 projected debit amount.
OFFL_20_PRJ_CDT	(number 21.3). Offline category #20 projected credit amount.
OFFL_20_PRJ_DBT	(number 21.3). Offline category #20 projected debit amount.
OFFL_21_PRJ_CDT	(number 21.3). Offline category #21 projected credit amount.
OFFL_21_PRJ_DBT	(number 21.3). Offline category #21 projected debit amount.
OFFL_22_PRJ_CDT	(number 21.3). Offline category #22 projected credit amount.
OFFL_22_PRJ_DBT	(number 21.3). Offline category #22 projected debit amount.
OFFL_23_PRJ_CDT	(number 21.3). Offline category #23 projected credit amount.
OFFL_23_PRJ_DBT	(number 21.3). Offline category #23 projected debit amount.
OFFL_24_PRJ_CDT	(number 21.3). Offline category #24 projected credit amount.
OFFL_24_PRJ_DBT	(number 21.3). Offline category #24 projected debit amount.
OFFL_25_PRJ_CDT	(number 21.3). Offline category #25 projected credit amount.
OFFL_25_PRJ_DBT	(number 21.3). Offline category #25 projected debit amount.
OFFL_26_PRJ_CDT	(number 21.3). Offline category #26 projected credit amount.
OFFL_26_PRJ_DBT	(number 21.3). Offline category #26 projected debit amount.
OFFL_27_PRJ_CDT	(number 21.3). Offline category #27 projected credit amount.
OFFL_27_PRJ_DBT	(number 21.3). Offline category #27 projected debit amount.

PSR Table (PSR_T)

OFFL_28_PRJ_CDT	(number 21.3). Offline category #28 projected credit amount.
OFFL_28_PRJ_DBT	(number 21.3). Offline category #28 projected debit amount.
OFFL_29_PRJ_CDT	(number 21.3). Offline category #29 projected credit amount.
OFFL_29_PRJ_DBT	(number 21.3). Offline category #29 projected debit amount.
OFFL_30_PRJ_CDT	(number 21.3). Offline category #30 projected credit amount.
OFFL_30_PRJ_DBT	(number 21.3). Offline category #30 projected debit amount.
OFFLINE_PRJ_CDT	(number 21.3). Offline category projected credit amount.
OFFLINE_PRJ_DBT	(number 21.3). Offline category projected credit amount.
OFFLINE_PRJ_NET	(number 21.3). Offline projected net amount.
THROTTLE_IND	(varchar 13). Indicates the state of the throttle.
EXTERNAL_LIMIT	(number 21.3). Maximum network overdraft limit, set by the bank.
INTERNAL_LIMIT	(number 21.3). Your bank's own internal debit cap. Provides a margin of safety between the network debit cap and the maximum amount you can send out via a network.
HOLD_THRESHOLD	(number 21.3). Optional limit your bank sets to hold individual transfers that exceed a specified amount, preventing their automatic release.
OPENNING_BALANCE	(number 21.3). Opening Fed balance, adjusted for purposes of overdraft monitoring.
POSITION_AMT	(number 21.3). Fed credits minus debits.
AVAIL_AMT	(number 21.3). Amount available before reaching your limit with the Fed.
STATEMENT_AS_OF	(number 21.3). End of the period covered by the statement.
STATEMENT_TIME	(date). Time at which the statement was generated.
RESERVE_TARGET	(number 21.3). Reserve target amount.
INPUTLOG_CNT	(number 10). Number of items on the input log.
INPUTLOG_AMT	(number 21.3). Total value of items on the input log.
OUTPUTLOG_CNT	(number 10). Number of items on the output log.
OUTPUTLOG_AMT	(number 21.3). Total value of items on the output log.
OPERATOR_ID	(varchar 10). The operator who initiated the change.

Relational Gateway Link Table (RGWLINK_T)

The Relational Gateway Link table is a processing log for the receiving side of message transmissions. It stores data for the RGW link between the sending side and the Oracle database. RGWLINK_T is used for Continuous RGW only.

The RGWLINK process loads RGWLINK_T, Message tables, and Message Processing Rules tables.

Primary Keys

The Relational Gateway Link table (RGWLINK_T) has the following primary keys:

- PROC_DATE
- LINKNAME
- TEIMSTAMP
- RECTYPE
- SERIAL

Column Descriptions

PROC_DATE (pk)	(char 8) Processing date from the MTS system.
LINKNAME (pk)	(varchar 16) Identifies the receiving link name that is assigned in the RGWLINK configuration file.
TIMESTAMP (pk)	(varchar 22) The date and time of this event. Format is:YYYY-MM-DD HH:MM:SS.CC
RECTYPE (pk)	(varchar 4) The event type, for example Message (MSG), Acknowledge (ACK), Resync (RSYN), Down (DOWN), or Up (UP).
SERIAL (pk)	(number 9) Sequence number, between 1 and 999,999, assigned to either a message (MSG) or to a message acknowledgement (ACK).
MESSAGE	(varchar 254) MTS TRN reference or other text message. Includes the date, transaction reference number, and serial number.

Remote Tables

The Remote tables provide information about connections to external systems and networks. The Remote tables contain link-specific parameters including transmission frame, line speed, communication protocol, time-out limit, and data format.

Parent/child relationships

The Remote table (REMOTE_T) is the parent table to the Remote Parameter table (REMOTE_PARAMETER_T).

Primary keys

The Remote tables have the following primary keys in common:

- RMT_IDBANK
- RMT_IDKEY
- RMT_TIME_STAMP

Remote Table (REMOTE_T)

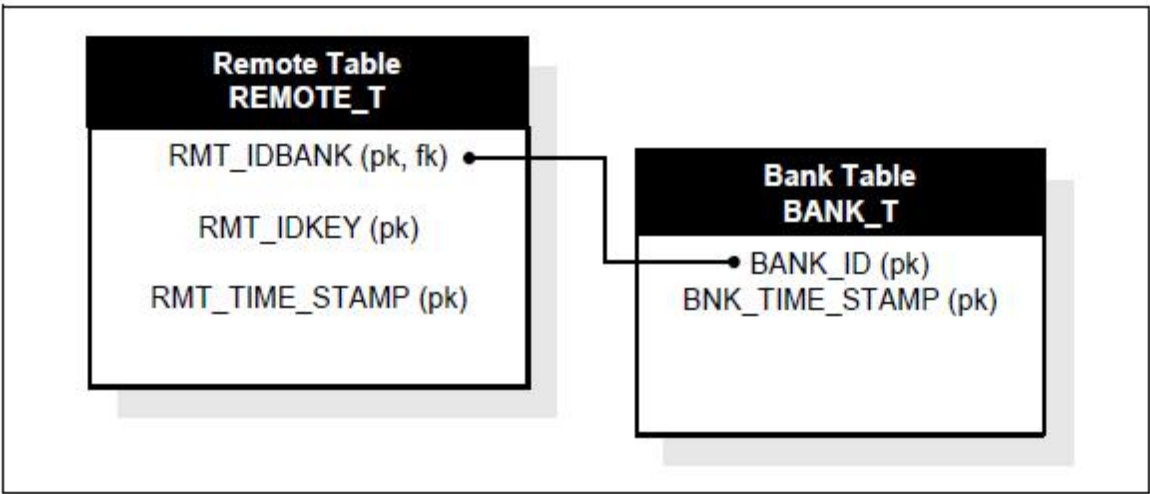
The Remote table describes link-specific parameters including transmission frame, line speed, communication protocol, time-out limit, and data format.

Table Relationships

The RMT_IDBANK column in the Remote table relates to the BANK_ID column in the Bank table. Additionally, the RMT_TIME_STAMP column in the Remote table relates to the BNK_TIME_STAMP column in the Bank table.

Relationship Diagram

This relationship diagram shows the Remote table along with the Operator and Bank tables.



Column Descriptions

RMT_IDBANK (pk, fk)	(varchar 3). Bank identifier of the owning bank; mapped from the RMT screen.
RMT_IDKEY (pk)	(varchar 20). Bank-specific remote system name; mapped from the RMT screen.
RMT_TIME_STAMP (pk)	(varchar 16). Date and time when the item was updated.
DAT_FUNCTION	(varchar 7). Function that last accessed the item in the Remote file; contains the following values: VFY ADD (Verify add)

Remote Tables

	VFY DEL (Verify delete) VFY UPD (Verify update)
CREATOR	(varchar 10). Operator ID and bank ID of the operator who added the item to the Remote file.
CREATE_DATE	(date). Date that the item was added to the Remote file.
UPDATER	(varchar 10). Operator ID and bank ID of the operator who updated the item in the Remote file.
UPDATE_DATE	(date). Date when the item was updated in the Remote file.
VERIFIER	(varchar 10). Operator ID and bank ID of the operator who verified the item in the Remote file.
VERIFY_DATE	(date). Date when the item was verified in the Remote file.
DAT_OWNER_PROD_ID	(varchar 3). Product ID.
DAT_OWNER_BANK_ID	(varchar 3). Bank ID.
DAT_OWNER_LOC	(varchar 6). Location ID.
DAT_OWNER_CUST	(varchar 20). Customer ID.
SYSTEM_NAME	(varchar 35). Name of the remote system.
CONTACT_NAME	(varchar 20). Name of the person to be contacted if a problem occurs (for example, if the communication line goes down). .
CONTACT_PHN	(varchar 20). Telephone number of the contact person. For a S.W.I.F.T. link using a dial-up modem, this column contains the autodial telephone number.
IN_LOGIN_ID	(varchar 20). Identifier that the remote system uses to log in to MTS.
IN_MIN_DAYS	(number 5). Minimum number of days the inbound password is valid; required when the inbound password is specified.
IN_MAX_DAYS	(number 5). Maximum number of days the inbound password is valid; required when the inbound password is specified.
IN_TEST_ALG	(varchar 6). Test used to validate access from the remote system.
IN_FIXED_NO	(number 10). Parameter for the inbound test algorithm.
OUT_LOGIN_ID	(varchar 20). Identifier MTS uses to log in to the outbound remote system.
OUT_MIN_DAYS	(number 5). Minimum number of days the outbound password is valid; required when the outbound password is specified.

OUT_MAX_DAYS	(number 5). Maximum number of days the outbound password is valid; required when the outbound password is specified.
OUT_TEST_ALG	(varchar 6). Test used to validate access from the outbound remote system.
OUT_FIXED_NO	(number 10). Parameter for the outbound test algorithm.
SOURCE_ID	(varchar 3). Mnemonic used to identify the sending system.
LINE_NAME	(varchar 12). For S.W.I.F.T. links, this field contains the name of the Login/Select Table; for all other links, the system uses this value to generate the name of the line log and other processing queues. (The line log contains information such as sequence numbers, time stamps of messages received and sent, and line status messages.)
LOCATION	(varchar 6). Location code associated with the remote system.
RELATED_IDTYPE	(varchar 1). Not used in this table.
RELATED_IDKEY	(varchar 64). Not used in this table.
REC_SIZE	(number 5). Physical record length; depends on the protocol. If unspecified, this column defaults to 80.
LOG_REC_SIZE	(number 5). Logical record length; depends on the protocol. If unspecified, this column defaults to 80.
PROT_TYPE	(varchar 10). Line-level protocol; the format of the data transmitted between sending and receiving systems.
LINE_SPEED	(number 5). Line speed.
BLOCK_SIZE	(number 5). Maximum transmission block or frame size; depends on the protocol.
SELECT_ID	(varchar 5). For S.W.I.F.T. links, this column contains the S.W.I.F.T. authentication algorithm mode. For bisynchronous links, this column contains the 3270 select address.
POLL_ID	(varchar 5). For S.W.I.F.T. links, this column contains the name of the detached process running S.W.I.F.T. line communication software. For bisynchronous links, this column contains the 3270 poll address.
DUPLEX_MODE	(varchar 4). Communication line characteristic; contains the following values: FULL (full duplex) or HALF (half duplex).
POLL_TIMEOUT	(number 5). Custom use only.
ACK_TIMEOUT	(number 5). Custom use only.
BURST_COUNT	(number 5). Number of records the link will send in one burst.
P1	(varchar 16). Parameter 1; specific to the link software.

Remote Tables

P2	(varchar 16). Parameter 2; specific to the link software.
P3	(varchar 16). Parameter 3; specific to the link software.
P4	(varchar 16). Parameter 4; specific to the link software.
P5	(varchar 16). Parameter 5; specific to the link software.
P6	(varchar 16). Parameter 6; specific to the link software.
P7	(varchar 16). Parameter 7; specific to the link software.
P8	(varchar 16). Parameter 8; specific to the link software.
TEST_FLAG	(varchar 1). Flag indicating whether the link is operating in test or production mode; contains the following values: Y (Normal test mode) N (Production mode) L (Loopback test mode; MTS returns the outbound messages to your bank and sends cable addresses to the bank name specified in the Bank file)
RESPONSE_FLAG	(varchar 1). Flag indicating whether the receiving system should send a response to report the status of the batch just received (Y) or not (N).
PACKED_REC	(varchar 1). Flag indicating whether batch transmission uses packed test records (Y) or not (N).
PEND_Q_TIMEOUT	(number 10). Pending queue time-out in seconds.
CURNT_LOGIN_TBL	(varchar 10). Current login table that MTS is using; contains the following values: NORMAL EMERGENCY FALLBACK MONITOR-X
TERM_ID	(varchar 12). Terminal ID.
APPL_ID	(varchar 10). Application ID.
CIRCUIT_ID	(varchar 10). Access address for S.W.I.F.T. links, specifies the remote network to use to make an outgoing call; contains the following values: DECPSIX.25 – access address from the NCP/NCL (network control program/network control language utilities) config. FREEWAY – name of the circuit as defined in the X25_FWAY_CONFIG_FILE parameter
RETRY_LIMIT	(number 5). Number of times the link will re-attempt to make a connection.
MODEM_PHN	(varchar 20). Modem phone number.

MODEM_PHN_FREQ	(varchar 12). Rate at which batches are to be transmitted; 12-character positional array of the form nnnnnnnnnnnn (no spaces). Each character position represents a two-hour time slot, beginning at midnight. Each character specifies the maximum number of batches that can be transmitted during each half hour in that time slot. For example, if the value is 111133331111, the first 1 represents the time slot from midnight until 2 am and specifies 1 batch per half hour, (or two batches per hour).
MASTER_SLAVE_FLAG	(varchar 1). Flag indicating whether MTS is master (M) or slave (S) in relationship with the remote system.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Remote Parameter Table (REMOTE_PARAMETER_T)

The Remote Parameter table includes the parameter ID, maximum size, and parameter value for each link parameter.

Table Relationships

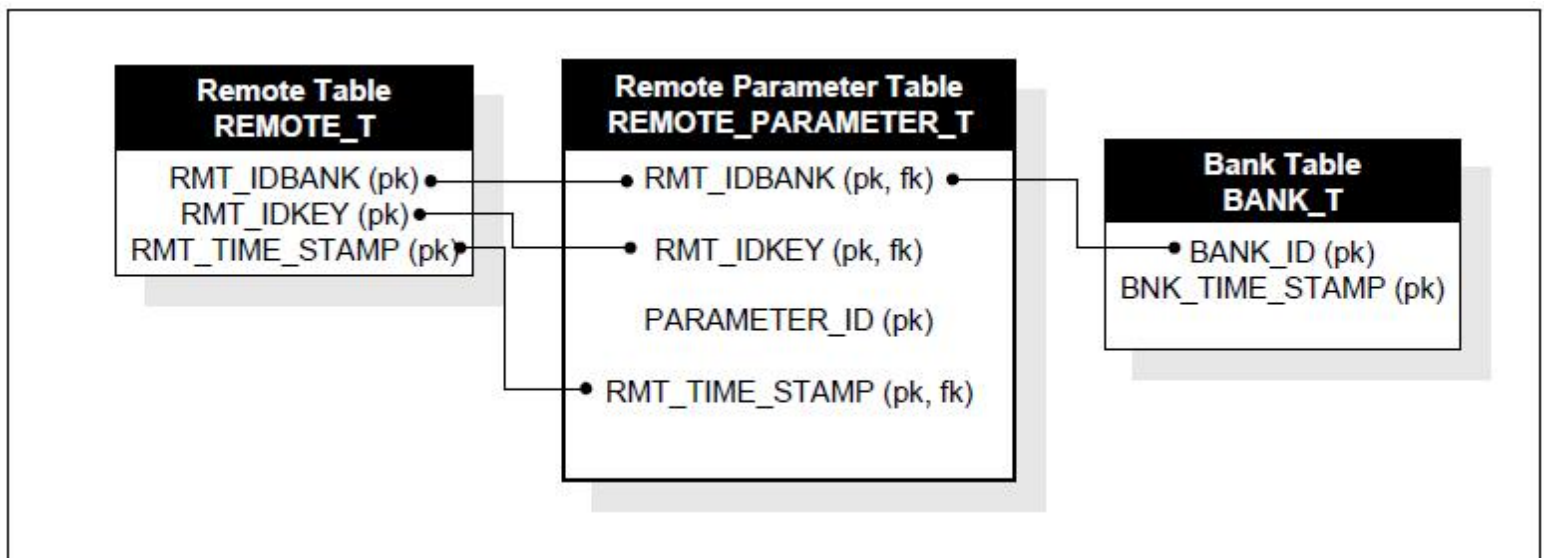
The following columns in the Remote table relate to the same columns in the Remote Parameter table:

- RMT_IDBANK
- RMT_IDKEY
- RMT_TIME_STAMP

The RMT_IDBANK column in the Remote Parameter table relates to the BANK_ID column in the Bank table. Additionally, the RMT_TIME_STAMP column in the Remote Parameter table relates to the BNK_TIME_STAMP column in the Bank table.

Relationship Diagram

This relationship diagram shows the Remote Parameter table and the Bank table.



Column Descriptions

RMT_IDBANK (pk, fk)	(varchar 3) Bank identifier of the bank that owns the remote line or resource; mapped from the RMT screen.
RMT_IDKEY (pk, fk)	(varchar 20) Bank-specific remote system name; mapped from the RMT selection screen.
PARAMETER_ID (pk)	(varchar 20) User-specified parameter name.
RMT_TIME_STAMP (pk, fk)	(varchar 16) Date and time when the item was updated.
MAXIMUM_SIZE	(number 5) Maximum size for the parameter.
PARAMETER_VALUE	(varchar 40) Value associated with a parameter name.
RECORD_EXPIRED	(varchar 16) Date and time when this item becomes invalid.
RECORD_UPDATED	(varchar 1) Indicates whether this row's timestamp column has been propagated to previous rows.

Rules Tables

Rules tables contain data about the rules used in the MTS function Rule Table Maintenance (RTM).

RTM provides access to rules that are applied to inbound SWIFT and telex messages. The rules, grouped together into categories called Rule Types, extract data to determine the message's final route. Data contained in the Rules tables includes: the static data used to identify messages and apply rules; From rule names; rule nicknames; generated rule patterns; and statistics of rule use. Rules tables include the following tables:

- Rule table (RULE_T)
- Rule From Name table (RULE_FROM_NAME_T)
- Rule Nickname table (RULE_NICKNAME_T)
- Rule Pattern table (RULE_PATTERN_T)
- Rule Statistics table (RULE_STATS_T)

Primary Keys

With the exception of the RULE_FROM_NAME_T table, the Rule tables have the following primary keys in common:

- RULE_BANK_ID
- RULE_NAME

Rule Table (RULE_T)

The Rule table stores the lists of rules used by the Rule Table Maintenance (RTM) function. Each rule contains static data used to identify messages and apply rules.

Primary Keys

The Rule table (RULE_T) table has the following primary keys:

- RULE_BANK_ID
- RULE_NAME
- RULE_TIME_STAMP

Column Descriptions

RULE_BANK_ID (pk)	(varchar 3). MTS bank code identifying the bank that owns the rule. A rule may be specific to a bank or may be generally available to routing and data extraction for all banks.
RULE_NAME (pk)	(varchar 30). Name of the rule. This is arbitrary and may be a descriptive identifier for the rule.
RULE_TIME_STAMP (pk)	(number 16). Day and time when the record was updated.
DAT_FUNCTION	(varchar 7). The function chosen: update, show, verify, or approve.
CREATOR	(varchar 10). User ID of the operator who created the record.
CREATE_DATE	(date). Date on which the record was created.
UPDATER	(varchar 10). User ID of the operator who updated the record.
UPDATE_DATE	(date). Date on which the record was updated.
VERIFIER	(varchar 10). User ID of the operator who verified the record.
VERIFY_DATE	(date). Date on which the record was verified.
DAT_OWNER_PROD_ID	(varchar 3). Product ID.
DAT_OWNER_BANK_ID	(varchar 3). Bank ID.
DAT_OWNER_LOC	(varchar 6). Location ID.

Rules Tables

DAT_OWNER_CUST	(varchar 20). Customer ID.
RULE_TYPE	(varchar 20). An arbitrary, possibly descriptive identifier for the rule.
RULE_PRIORITY	(number 10). Defines the order in which the rule is executed within the rule type. Range is from 1 to 999999.
RULE_WIRE_TYPE	(varchar 10). Rule wire type.
RULE_SCOPE	<p>(varchar 10). Defines whether the execution of the rule terminates after the first match or whether it is repeated through the full source text. Most rule types have a set scope, specific to the processing performed for the rules of that type. The scope can be either of the following:</p> <p>FIRST. Only the first match made fills the Target with the extracted data.</p> <p>FULL. Multiple matches can be made, filling the Target with the concatenated extracted data from each match. Within the Target, an end-of-line character delineates the data extracted from each match.</p>
SOURCE_TYPE	<p>(varchar 10). Specifies the type of source text. Options are:</p> <p>MESSAGE (the entire message)</p> <p>ROUTES (the candidate route list)</p> <p>FIELD (a specified message field)</p> <p>MEMO (specified tagged data to be secured as a memo in a message history segment)</p> <p>TEMP (a specified element of Searcher temporary storage)</p>
SOURCE_NAME	(varchar 30). Name of the field or data storage
TARGET_TYPE	<p>(varchar 10). The type of target. Options are:</p> <p>FIELD. A specified message field.</p> <p>MEMO. Specified tagged data to be secured as a memo in a message history segment.</p> <p>TEMP. A specified element of Searcher temporary storage.</p> <p>BOOLEAN. A named flag value.</p>
TARGET_NAME	(varchar 30). The name of the specific target.
FIRST_NUM	(number 10). First number.
LAST_NUM	(number 10). Last number.
RULE_ROUTEQ_PROD_ID	(varchar 3). Routing queue product ID.
RULE_ROUTEQ_BANK_ID	(varchar 3). Routing queue bank ID.
RULE_ROUTEQ_LOC	(varchar 6). Routing queue location ID.
RULE_ROUTEQ_CUST	(varchar 20). Routing queue customer name.

RULE_ROUTEQ_NAME	(varchar 33). Routing queue name.
FINAL_RULE	(char 1). Used only for Simple Routing rules. If set to Y, the route is determined and no other Simple Routing rules are executed. If set to N (default) the matched rule is added to the Candidate Route List.
RESULT	(varchar 35). Identifies the data to be moved into the Target string when the regular expression matches. The result can contain either of the following: A constant string, allowing the Target field to be set to a specific value when the regular expression matches. \$nn, specifying that the Target field is to be set to an extracted portion of the source string, where nn identifies the number of the parentheses set group with the regular expression.
NAME_RULE	(char 1). Indicates whether a full rule name is required to select that rule. Partial entry of the rule name limits the summary list to all rules matching the entry.
FIRST_NAME	(varchar 35). Optional first name. If present, the regular expression will be generated for the full first name or its initial (first character), with or without a terminator (period). .
MIDDLE_NAME	(varchar 35). Optional middle name. If present, the regular expression will be generated for the full name or its initial (first character), with or without a terminator (period).
LAST_NAME	(varchar 35). Required. The regular expression will be generated for an exact match on the full last name.
RULE_APPROVED	(char 1). Indicates whether the verified rule has been approved for production use.
RULE_APPROVER	(varchar 35). User ID of the operator who approved the rule for production use.
RULE_APPROVE_TIME	(date). Date on which the rule was approved for production use.
RULE_ACTIVATE_TIME	(date). Date and time on which the rule is to be activated for use in matching.
RULE_DEACTIVATE_TIME	(date). Date and time on which the rule is to be deactivated and no longer used in matching.
RECORD_EXPIRED	(number 16). Date and time when this item becomes invalid.
RECORD_UPDATED	(char 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Rule From Name Table (RULE_FROM_NAME_T)

The Rule From Name table stores the lists of From names used by the Rule Table Maintenance (RTM) function.

FROM rules are applied to messages to derive the potential name and location of the originator. The name is validated against the REL FROM index sequence to obtain the originator's REL ID.

Primary Keys

The Rule From Name table (RULE_FROM_NAME_T) table has the following primary keys:

- FROM_NAME
- FROM_NAME_TIME_STAMP

Column Descriptions

FROM_NAME (pk)	(varchar 80) Name of the FROM rule.
FROM_NAME_TIME_STAMP (pk)	(number 16) Date and time when the record was last updated.
CREATOR	(varchar 10) User ID of the operator who created the record.
CREATE_TIME	(date) Date on which the record was created.
DAT_OWNER_PROD_ID	(varchar 3) Product ID.
DAT_OWNER_BANK_ID	(varchar 3) Bank ID.
DAT_OWNER_LOC	(varchar 6) Location ID.
DAT_OWNER_CUST	(varchar 20) Customer ID.
REL_ID	(number 10) Unique reference number that identifies this relationship.
LOG_MEMO	(varchar 80) Memo from log.
INDEX_MEMO	(varchar 80) Memo from index.
RECORD_EXPIRED	(number 16) Date and time when this item becomes invalid.

RECORD_UPDATED	(char 1) Indicates whether this row's timestamp column has been propagated to previous rows.
----------------	--

Rule Nickname Table (RULE_NICKNAME_T)

The Rule Nickname table stores the lists of nicknames associated with a rule used in the Rule Table Maintenance (RTM) function.

Nicknames, which replace the first and middle names given to a message, may be included in a message. Up to five nicknames can be specified for a single message.

Primary Keys

The Rule Nickname table (RULE_NICKNAME_T) table has the following primary keys:

- RULE_BANK_ID
- RULE_NAME
- NICKNAME_SEQ_NUM
- NICKNAME_TIME_STAMP

Column Descriptions

RULE_BANK_ID (pk)	(varchar 3). MTS bank code identifying the bank that owns the rule. A rule may be specific to a bank or may be generally available to routing and data extraction for all banks.
RULE_NAME (pk)	(varchar 30). An arbitrary, possibly descriptive identifier for the rule.
NICKNAME_SEQ_NUM (pk)	(number 10). Rule sequence number
NICKNAME_TIME_STAMP (pk)	(number 16). Date and time when the record was updated.
NICKNAME	(varchar 35). Name that identifies the rule. The nickname replaces both the first and middle names.
RECORD_EXPIRED	(number 16). Date and time when this item becomes invalid.
RECORD_UPDATED	(char 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Rule Pattern Table (RULE_PATTERN_T)

The Rule Pattern table stores the generated rules patterns used by the Rule Table Maintenance (RTM) function.

For Complex Routing rules, the pattern data is a number of Boolean rule names combined with AND logic. The NOT of a Boolean rule is identified by the rule name being preceded by a minus sign. If all the rules (or their negation) evaluate to TRUE, the route is selected. For all other rule types, the pattern data is a regular expression that captures data, determines a route, or sets a flag.

Primary Keys

The Rule Patter table (RULE_PATTERN_T) has the following primary keys:

- RULE_BANK_ID
- RULE_NAME
- PATTERN_SEQ_NUM
- PATTERN_TIME_STAMP

Column Descriptions

RULE_BANK_ID (pk)	(varchar 3). MTS bank code identifying the bank that owns the rule. A rule may be specific to a bank or may be generally available to routing and data extraction for all banks.
RULE_NAME (pk)	(varchar 30). An arbitrary, possibly descriptive identifier for the rule.
PATTERN_SEQ_NUM (pk)	(number 10). Rule sequence number.
PATTERN_TIME_STAMP (pk)	(number 16). Date and time when the pattern's record was updated.
GENERATED_PATTERN	(varchar 80). The system-generated regular expression.
RECORD_EXPIRED	(number 16). Date and time when this item becomes invalid.
RECORD_UPDATED	(char 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Rule Statistics Table (RULE_STATS_T)

The Rule Statistics table stores statistics about rule matching that has occurred. Statistics include the number of messages against which the rule was checked and the number of messages against which the rule was matched.

Primary Keys

The Rule Statistics table (RULE_STATS_T) has the following primary keys:

- RULE_BANK_ID
- RULE_NAME
- STATS_SEQ_NUM
- STATS_TIME_STAMP

Column Descriptions

RULE_BANK_ID (pk)	(varchar 3). MTS bank code identifying the bank that owns the rule. A rule may be specific to a bank or may be generally available to routing and data extraction for all banks.
RULE_NAME (pk)	(varchar 30). An arbitrary, possibly descriptive identifier for the rule.
STATS_SEQ_NUM (pk)	(number 10). Rule sequence number.
STATS_TIME_STAMP (pk)	(number 16). Date and time when the record was updated.
TIME_LOGGED	(number 16). Time when the searched logged the rule.
SEARCHER_NAME	(varchar 80). Name of the server that tracked the performance.
CHECK_COUNT	(number 10). Number of messages against which the rule was checked.
MATCH_COUNT	(number 10). Number of messages against which the rule was matched.
RECORD_EXPIRED	(number 16). Date and time when this item becomes invalid.
RECORD_UPDATED	(char 1). Indicates whether this row's timestamp column has been propagated to previous rows.

SEPA Message Tables (MTS 4.0 and Higher)

This topic lists Message tables available for SEPA-enabled systems.

This document describes the following SEPA message tables provided as part of the MTS SEPA functionality:

- Message ACH Table (MESSAGE_ACH_T)
- Message ACH Regulatory Reporting Table (MESSAGE_ACH_RRPT_T)
- Message BCM Table (MESSAGE_BCM_T)
- Message Batch Control Messages Item Table (MESSAGE_BCM_ITEM_T)
- Message Batch Control Messages Fund Table (MESSAGE_BCM_FUND_T)

The Message Table (MESSAGE_T) is parent to the MESSAGE_ACH_T and MESSAGE_BCM_T tables. These tables are in turn parents to their similarly-named child tables.

Primary Keys

The SEPA message tables have the ROW_ID key in common with their parent, the MESSAGE_T table.

The SEPA message tables share the foreign key PARENT_ID.

Message ACH Regulatory Reporting Table (MESSAGE_ACH_RRPT_T)

This table contains regulatory reporting information for SEPA messages.

Column Descriptions

Column	Format	Description
Row_id	Number(16) NOT NULL	Unique, Oracle assigned id for the table row.
Parent_id	NUMBER(16)NOT NULL	Row_id of the parent in MESSAGE_ACH_T'.
DbtCdtRptgInd	VARCHAR2(4)	Debit/credit indicator. Defines Side for reporting BOTH, CRED, DEBT.
AuthrtyNm	VARCHAR2(70)	Name of Entity requiring regulatory reporting information.
AuthrtyCtry	VARCHAR2(2)	ISO Country code of Entity requiring regulatory reporting information.
RgltryDtls_Cd	VARCHAR2(3)	Reason for transaction regualtory reporting requirement.
RgltryDtls_Amt	NUMBER(21,3)	Amount for regulatory reporting requirements.
RgltryDtls_Ccy	VARCHAR2(3)	Currency of amount.
RgltryDtls_Inf	VARCHAR2(35)	Additional Details for specific regulatory requirements.

Message ACH Table (MESSAGE_ACH_T)

The Message ACH table contains information about SEPA messages.

Column Descriptions

Column	Format	Description
Row_id	NUMBER(16) NOT NULL	Unique, Oracle assigned id for the table row
Parent_id	NUMBER(16) NOT NULL	Row_id of the parent in MESSAGE_T
input_format	VARCHAR2(10)	MTS input format name (EBASCTICF, ISOPAIN001, ISOPACS003, etc.)
output_format	VARCHAR2(10)	MTS output format name.
PCM_trn_ref	VARCHAR2(16)	Physical file control message transaction reference, EBA Bundle.
FCM_trn_ref	VARCHAR2(16)	Logical file control message transaction reference, PAIN Single and Mixed.
BCM_trn_ref	VARCHAR2(16)	Batch control message transaction reference, PACS and PAIN Grouped.
FwdgAgt_flg	VARCHAR2(20)	Forwarding Agent flag.
PmtInfid_flg	VARCHAR2(20)	Payment Method info flag.
pmt_mtd	VARCHAR2(3)	Payment Method ISO PAIN, CHK - check, TRA transfer w/advice, TRF transfer.
Instr_id	VARCHAR2(35)	Instr_id IS Instruction ID from instruction party.
Endtoend_id	VARCHAR2(35)	End to end ID from initiating Party.
Tx_id	VARCHAR2(35)	Transaction ID assigned by first instructing agent in the originating financial institution.
Pmt_tp_inf_flg	VARCHAR2(20)	Payment type information flag.
Instr_prty_flg	VARCHAR2(20)	Instruction Priority flag.
Svclvl_tp	VARCHAR2(1)	Service Level type. "C" for Code or "P" for Proprietary. Must be "C" for SEPA.
Svclvl	VARCHAR2(35)	Service Level. "SEPA" for SEPA.
Lclinstrm_tp	VARCHAR2(1)	Local Instrument type. "C" for Code or "P" for proprietary.
Lclinstrm	VARCHAR2(35)	Local Instrument. When type is "C," it is four-character code, When "P", the value is up to 35 characters.
Seq_tp	VARCHAR2(4)	SDD sequence type OOFF, FRST, RCUR or FNAL.
Ctgy_purp	VARCHAR2(4)	Category Purpose flag. Valid value defined by ISO.
Reqd_date	DATE	Original requested date, SCT req exec date, or SDD req Colleciton date. YYMMDD.
IntrBkSettlm_date	DATE	PACS interbank settlement date. YYMMDD.
Sttlm_tm_indctn_flg	VARCHAR2(20)	Settlement Time Ind.

SEPA Message Tables (MTS 4.0 and Higher)

Column	Format	Description
Sttlm_tm_req_flg	VARCHAR2(20)	Settlement Time Request.
Acptnc_dt_tm_flg	VARCHAR2(20)	Acceptance Date time flag.
Poolg_adjstmnt_dt_flg	VARCHAR2(20)	Pooling Adjustment Flag.
Instd_amt_flg	VARCHAR2(20)	Instructed Amount Flag.
Eqvt_amt_flg	VARCHAR2(20)	Equivalent Amount Flag.
Xchg_rate_inf_flg	VARCHAR2(20)	Exchange Info Rate Flag. ISO PAIN SCT Only.
Xchg_rate_flg	VARCHAR2(20)	Exchange Rate Flag.
Chrgs_inf_flg	VARCHAR2(20)	Charges Info Flag.
Plnf_Chrgacct_flg	VARCHAR2(20)	Payment Info/Charges Acct Flag.
Plnf_Chrgacct_agt_flg	VARCHAR2(20)	Payment info/charges Acct Agent.
Prvs_instg_agt_flg	VARCHAR2(20)	Previous Instructing Agent.
Prvs_instg_acct_flg	VARCHAR2(20)	Previous Instructing Agent Account.
Instg_agt	VARCHAR2(35)	Instructing Agent BIC for SEPA.
Instd_agt	VARCHAR2(35)	Instructed Agent BIC for SEPA.
Intrmy_agt1_flg	VARCHAR2(20)	Intermediary Agent 1 flag.
Intrmy_agt1acct_flg	VARCHAR2(20)	Intermediary Agent 1 account flag.
Intrmy_agt2_flg	VARCHAR2(20)	Intermediary Agent 2 flag.
Intrmy_agt2acct_flg	VARCHAR2(20)	Intermediary Agent 2 account flag.
Intrmy_agt3_flg	VARCHAR2(20)	Intermediary Agent 3 flag.
Intrmy_agt3acct_flg	VARCHAR2(20)	Intermediary Agent 3 account flag.
ChqInstr_flg	VARCHAR2(20)	Cheque Instruction flag. ISO and EPC SCT only. Contains Extensive Check Data.
Initg_pty_flg	VARCHAR2(20)	Initiating Party Flag.
Dbtr_agtacct_flg	VARCHAR2(20)	Debtor Agent Acct Flag.
Cdtr_agtacct_flg	VARCHAR2(20)	Creditor Agent Acct Flag.
Instr_cdtr_agt_cd	VARCHAR2(4)	Instruction for Creditor Agent code.
Instr_nxt_agt_flg	VARCHAR2(20)	Instr for Next Agents Flag.
Instr_dbtr_agt_flg	VARCHAR2(20)	Instr for Debtor Agent Flag.
Purp_tp	VARCHAR2(1)	Purpose type. "C" for Code, "P" for Proprietary.
Purp	VARCHAR2(35)	Purpose. When type is C, this is the maximum 4 character ISO code. When type is P, this is the maximum 35 character ISO code.
Tax_flg	VARCHAR2(20)	Tax Flag.
Mndt_id	VARCHAR2(35)	Mandate Id.
Dt_of_Sgntr	DATE	Date of Signature - YYYYMMDD.
Amdmnt_Ind	VARCHAR2(1)	Amendment indicator. "Y" = amendment data is provided.
Orgnl_mndtid	VARCHAR2(35)	Original Mandate Id.

SEPA Message Tables (MTS 4.0 and Higher)

Column	Format	Description
Orgnl_Fnl_Colltn_Dt	DATE	Original Final Collection Date.
Orgnl_Frqcy	VARCHAR2(4)	Original frequency: YEAR, MNTH, WEEK, QURT, etc.
Elctrnc_Sgntr	VARCHAR2(20)	Electronic Signature flag.
Frst_Colltn_Dt	DATE	First Collection Date YYMMDD.
Fnl_Colltn_Dt	DATE	Final Collection Date YYMMDD.
Frqcy	VARCHAR2(4)	Frequency: YEAR, MNTH, WEEK, QURT, etc.
PreNtfctn_Id	VARCHAR2(35)	Pre Notification Identification.
PreNtfctn_Dt	DATE	Pre Notification Date.
Input_exc_reason_tp	VARCHAR2(1)	Input exception reason type. C for code. P for Proprietary.
Input_exc_reason	VARCHAR2(35)	Input exception reason. When type is C, value is ISO code (4 maximum). When P, the maximum is 35.
Output_exc_reason_tp	VARCHAR2(1)	Output exception reason type. C for code. P for Proprietary.
Output_exc_reason	VARCHAR2(35)	Output exception reason. When type is C, value is ISO code (4 maximum). When type is P, the maximum is 35.
Orgnl_MsgId	VARCHAR2(35)	Original Message ID.
Orgnl_MsgNmId	VARCHAR2(35)	Original message name identification, original pacs in a pacs 002/004/007.
Grp_Sts	VARCHAR2(4)	Group status RJCT, etc.
Tx_Sts	VARCHAR2(4)	Transaction Status.
Orgnl_Instr_Id	VARCHAR2(35)	Original Instruction ID.
Orgnl_EndToEnd_Id	VARCHAR2(35)	Original end-to-end ID.
Orgnl_Tx_Id	VARCHAR2(35)	Original transaction ID.
Orgnl_IntrBkSttlm_Dt	DATE	Original Interbank Settlement Date.
Orgnl_IntrBkSttlm_Amt	NUMBER(21,3)	Original Interbank Settlement Amount.
Orgnl_IntrBkSttlm_Ccy	VARCHAR2(3)	Original Interbank Settlement Amount currency.
Compstn_amt_flg	VARCHAR2(20)	Compensation Amount Flag.
Compstn_ind	VARCHAR2(2)	Compensation Indicator.
gen_bulk_excp_flg	VARCHAR2(1)	Generated bulk exception, set by bulk exception on transactions that it creates.
Input_File_Id	VARCHAR2(35)	File Header ID in the ACH Input File.
Output_File_Id	VARCHAR2(35)	File Header ID in the ACH Output File.
Input_File_Nm	VARCHAR2(35)	The network name of the ACH Input File.
Output_File_Nm	VARCHAR2(35)	The network name of the ACH Output File.
Input_MsgId	VARCHAR2(35)	The Message Identification <MsgId> of the PACS or PAIN input file that contained this transaction.
Output_MsgId	VARCHAR2(35)	Value placed in the <MsgId> field of the outbound PACS file containing this transaction. Also placed in

SEPA Message Tables (MTS 4.0 and Higher)

Column	Format	Description
		the Settlement Transaction created for the outbound file.
Output_Tx_Id	VARCHAR2(35)	Value placed in the the <TxId>, Status Identificaton <StsId>, Return Identification <RtnId> or Reversal Identification <RvsId> of the outbound PACS file containing this transaction.
Output_Instr_Id	VARCHAR2(35)	Value placed in the Instruction Identification <InstId> field of the outbound PACS file containing this transaction.
Instr_cdtr_agt_inf	VARCHAR2(140)	First Instruction for Creditor Agent.
Instr_cdtr_agt_flg	VARCHAR2(20)	Instruction for CDTR Agents Flag.
ISO_Dbtr_acct	VARCHAR2(4000)	Identification of the account to which a debit entry will be made as a result of the transaction.
ISO_Dbtr_agt	VARCHAR2(4000)	Financial institution servicing an account for the debtor.
ISO_Dbtr_pty	VARCHAR2(4000)	Party that owes an amount of money to the ultimate creditor.
ISO_Ultmt_Dbtr	VARCHAR2(4000)	Ultimate Party that owes an amount of money to the ultimate creditor.
ISO_Cdtr_acct	VARCHAR2(4000)	Identification of the account to which a credit entry will be posted as a result of the transaction.
ISO_Cdtr_agt	VARCHAR2(4000)	Financial institution servicing an account for the creditor.
ISO_Cdtr_pty	VARCHAR2(4000)	Identification of the account of the creditor agent at its servicing agent.
ISO_Ultmt_Cdtr	VARCHAR2(4000)	Ultimate Party to which an amount of money is due.
ISO_OrgnlCdtrSchmId	VARCHAR2(4000)	Original creditor scheme identification that has been modified.
ISO_OrgnlCdtrAgt	VARCHAR2(4000)	Original creditor agent that has been modified.
ISO_OrgnlCdtrAgtAcct	VARCHAR2(4000)	Original creditor agent account that has been modified.
ISO_OrgnlDbtr	VARCHAR2(4000)	Original debtor that has been modified.
ISO_OrgnlDbtrAcct	VARCHAR2(4000)	Original debtor account that has been modified.
ISO_OrgnlDbtrAgt	VARCHAR2(4000)	Original debtor's agent that has been modified.
ISO_OrgnlDbtrAgtAcct	VARCHAR2(4000)	Original debtor agent account that has been modified.
ISO_CdtSchme_id	VARCHAR2(4000)	Credit party that signs the direct debit mandate.
ISO_Exc_Orgtr	VARCHAR2(4000)	Exception originator.
XML_schema_Id	NUMBER(16)	Physical file control message transaction reference, EBA Bundle.

Message Batch Control Messages Fund Table (MESSAGE_BCM_FUND_T)

This table contains fund information about batch control messages.

Column Description

Column	Format	Description
Row_id	NUMBER(16) NOT NULL	Unique, Oracle assigned id for the table row.
Parent_id	NUMBER(16) NOT NULL	Row_id of the parent in MESSAGE_BCM_T.
Fund_id	VARCHAR2(8)	Fund Identification.
Dda_num	VARCHAR2(30)	DDA Account.
Amt	NUMBER(21,3)	Total Amount.
Currency_code	VARCHAR2(3)	Currency code.
Beg_itm	VARCHAR2(11)	First item in the range.
End_itm	VARCHAR2(11)	Last item in the range.

Message Batch Control Messages Item Table (MESSAGE_BCM_ITEM_T)

This table contains information about the creation and verification of the items in the batch control messages.

Column Description

Column	Format	Description
Row_id	NUMBER(16) NOT NULL	Unique, Oracle assigned id for the table row
Parent_id	NUMBER(16) NOT NULL	Row_id of the parent in MESSAGE_BCM_T
Beg_itm	VARCHAR2(11)	The first item in the batch.
End_itm	VARCHAR2(11)	The last item in the batch.
Rng_sts	VARCHAR2(20)	Definition of possible status values for an item range in the item_range_seq of the batch control message.
Cre_opr	VARCHAR2(10)	Operator that created the batch.
Cre_date	DATE	Time created.
Vfy_opr	VARCHAR2(10)	Operator that verified the batch.
Vfy_date	DATE	Time verified.

Message BCM Table (MESSAGE_BCM_T)

This table contains information about batches control messages.

Column Descriptions

Column	Format	Description
Row_id	NUMBER(16) NOT NULL,	Unique, Oracle assigned id for the table row.
Parent_id	NUMBER(16) NOT NULL	Row_id of the parent in MESSAGE_T.
Batch_id_date	VARCHAR2(6)	Date section of batch id.
Batch_id_num	VARCHAR2(4)	Sequential number of batch id.
Eirn_date	VARCHAR2(5)	The date of the document created in Payment Entry.
Eirn_bat_num	VARCHAR2(7)	The batch number of the document created in Payment Entry.
Batch_status	VARCHAR2(20)	Definition of status for a Batch control message.
Date_error	VARCHAR2(1)	Bcm_flag. Set when received batch header has bad date.
Sequence_error	VARCHAR2(1)	Bcm_flag. Set when received batch header has bad sequence number.
Job_name_error	VARCHAR2(1)	Bcm_flag. Set when received header has bad job name.
Password_error	VARCHAR2(1)	Bcm_flag. Set when received header has bad password.
Totals_error	VARCHAR2(1)	Bcm_flag, When trailer totals not equal to our totals.
Testkey_error	VARCHAR2(1)	Bcm_flag, Set when header test key is invalid.
Incomplete_error	VARCHAR2(1)	Bcm_flag. Set when batch is incomplete/aborted.
Tot_item_amt	NUMBER(21,3)	Cumulative item value calculated by input mapper.
Tot_item_cnt	VARCHAR2(11)	Number of messages within batch as seen by input mapper.
Fed_amt	NUMBER(21,3)	Cumulative item value of all FED messages for this batch.
Fed_cnt	VARCHAR2(11)	Number of messages in batch for which FED msgs have been created.
Nonfed_amt	NUMBER(21,3)	Cumulative item value of all non-FED messages for this batch.
Nonfed_cnt	VARCHAR2(11)	Number of messages in batch for which non-FED messages have been created.
Trlr_amt	NUMBER(21,3)	Cumulative item value from the trailer record.
Trlr_cnt	VARCHAR2(11)	Number of messages in batch from the trailer record.
Pend_acc_cnt	VARCHAR2(11)	Number of items awaiting acceptance.
Pend_acc_amt	NUMBER(21,3)	Value of items awaiting acceptance.
Pend_vfy_cnt	VARCHAR2(11)	Number of items awaiting verification.

SEPA Message Tables (MTS 4.0 and Higher)

Column	Format	Description
Pend_vfy_amt	NUMBER(21,3)	Value of items awaiting verification.
Num_processed_in_fees	VARCHAR2(11)	Number of messages processed in fees.
Number_cancelled	VARCHAR2(11)	Number of messages cancelled.
Number_in_batch	VARCHAR2(11)	Passed fees and cancelled.
Number_verified	VARCHAR2(11)	Independent from other three counters.
Duplicate_error	VARCHAR2(1)	Set when duplicate batch is received.

Mandate Table (MANDATE_T)

The Mandate table contains information about SEPA direct debit mandate messages.

Column Descriptions

Column	Format	Description
Creditor_id (PK)	VARCHAR2(35) NOT NULL	Credit party that signs the direct debit mandate.
Mandate_id (PK)	VARCHAR2(35) NOT NULL	Reference of the direct debit mandate has been signed between the debtor and creditor .
Mandate_Time_Stamp (PK)	VARCHAR2(16) NOT NULL	Unique Date/time stamp when row created.
Seq_tp	VARCHAR2(4)	SDD sequence type OOFF, FRST, RCUR or FNAL.
Dt_of_Sgntr	DATE	Date of Signature - YYMMDD.
Frst_Colltn_Dt	DATE	Date of First Collection - YYMMDD.
FnI_Colltn_Dt	DATE	Date of Final Collection - YYMMDD.
ISO_Dbtr_pty	VARCHAR2(4000)	Party that owes an amount of money to the ultimate creditor.
ISO_Dbtr_acct	VARCHAR2(4000)	Identification of the account to which a debit entry will be made as a result of the transaction.
ISO_Dbtr_agt	VARCHAR2(4000)	Financial institution servicing an account for the debtor.
ISO_Cdtr_pty	VARCHAR2(4000)	Identification of the account of the creditor agent at its servicing agent.
ISO_Cdtr_agt	VARCHAR2(4000)	Financial institution servicing an account for the creditor.
ISO_CdtSchme_id	VARCHAR2(4000)	Credit party that signs the direct debit mandate.
PreNtfctn_Id	VARCHAR2(35)	Pre Notification Identification.
PreNtfctn_Dt	DATE	Pre Notification Date.
Amendment_trn	VARCHAR2(16)	Final transaction for direct debit.
Fnal_trn	VARCHAR2(16)	Date/Time that this record becomes invalid.
Record_Expired	VARCHAR2(16)	Whether this row has had its timestamp propagated.
Record_Updated	VARCHAR2(1) DEFAULT 'N' NOT NULL	Flag to identify if record is current "Y" or deleted "N" but not physically removed.

Standing Order Tables

The Standing Order Tables contain information that MTS uses for initiating standing orders and for generating the STO Activity report.

Primary keys

The standing order tables have no primary keys in common.

Standing Order Table (STANDING_ORDER_T)

The Standing Order table provides information about the parameters MTS uses to initiate standing orders.

For each standing order, you identify the preexisting repetitive template that MTS will use to generate the payment, the type and amount of the transfer, and the payment schedule.

Table Relationships

The STO_BANK, STO_TYPE, and STO_KEY_ACC columns in the Standing Order table relate to the BANK_ID, ID_TYPE, and ACCOUNT columns in the Account table.

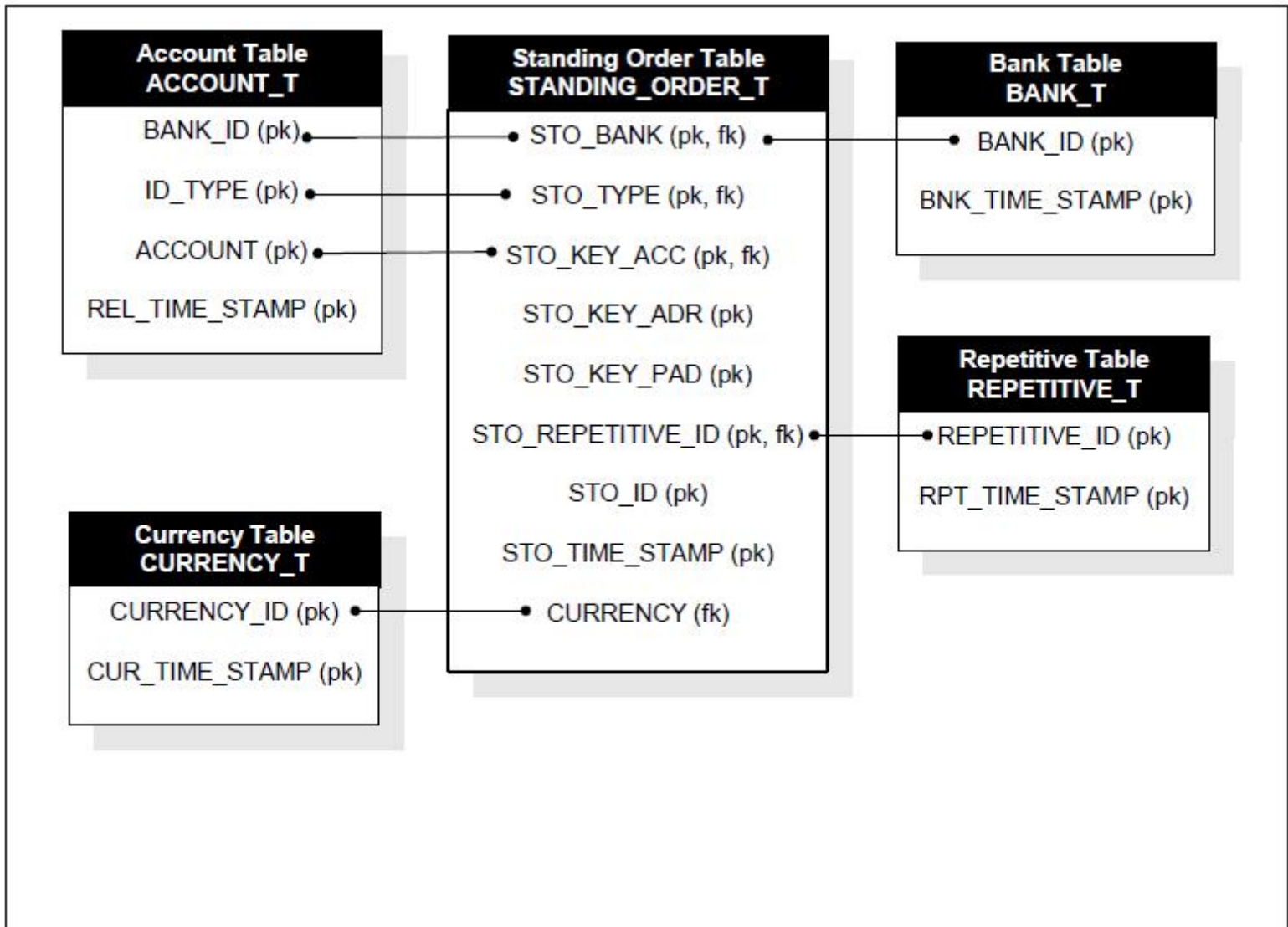
The STO_BANK column in the Standing Order table relates to the BANK_ID column in the Bank table.

The STO_REPETITIVE_ID column in the Standing Order table relates to the REPETITIVE_ID column in the Repetitive table.

The CURRENCY column in the Standing Order table relates to the CURRENCY_ID column in the Currency table.

Relationship Diagram

This relationship diagram shows the Standing Order table and other tables related to it through foreign keys.



Column Descriptions

STO_BANK (pk, fk)	(varchar 3). Bank identifier of the owning bank.
STO_TYPE (pk, fk)	(varchar 1). Account ID type.
STO_KEY_ACC (pk, fk)	(varchar 30). Account number used to find an address.

STO_KEY_ADR (pk)	(varchar 5). Address code used to distinguish between multiple addresses sharing the same account.
STO_KEY_PAD (pk)	(varchar 1). Number of additional lines for an address when there are several addresses.
STO_REPETITIVE_ID (pk, fk)	(varchar 8). ID of the repetitive (from the repetitive file). to be used to initiate this standing order.
STO_ID (pk)	(varchar 6). Sequential identifier MTS assigned to this standing order when it was created.
STO_TIME_STAMP (pk)	(varchar 16). Date and time when the standing order was created or updated.
TYPE	(varchar 6). Type of standing order; contains the following values: BALPEG (Draw down the account to a peg balance when the account balance reaches the amount in the STO UPPER LIMIT field on the STO Detail screen.) FIXED (Debit the account for a fixed amount) PEG (Draw down the account to a peg balance using the account's available balance) PEG_L (Draw account down to a peg balance using the ledger balance)
AMOUNT	(number 21.3). If the STO type is PEG, BALPEG, or PEG_L, this column contains the amount of the peg balance (this value is zero if the opening balance is to be drawn down at the start of a day). If the STO type is FIXED, this column contains the amount to be debited.
CURRENCY (fk)	(varchar 3). Currency involved in the standing order.
UPPER_LIMIT	(number 21.3). Account balance that triggers a BALPEG standing order.
INCREMENT_AMOUNT	(number 21.3). Increment by which MTS reduces an account balance when executing a PEG, PEG-L, or BALPEG standing order. For example, an account has a standing order with an STO Amount equal to \$5,000 and an Increment Amount equal to \$1,000. When MTS executes this STO, the account balance is reduced in \$1,000 increments until the STO amount is reached. If the account has a current balance equal to \$42,250 in the current example, MTS reduces the account value to \$5,250 when the standing order executes.

FREQUENCY	(varchar 12). Frequency of the standing order; contains the following values: DAILY WEEKLY BI-WEEKLY MONTHLY END-OF-MONTH OTHER
GRID	(varchar 7). Not populated in this table.
BEGINS_TIME	(date). Initial processing date for this standing order.

Standing Order Tables

EXPIRES_TIME	(date). Expiration date for this standing order.
PREVIOUS_TIME	(date). Date when this standing order was last used.
NEXT_TIME	(date). Next scheduled processing date for this standing order.
TIME_1	(date). User-specified timestamp 1; used when the FREQUENCY type is OTHER to indicate when the standing order is performed.
TIME_2	(date). User-specified timestamp 2; used when the FREQUENCY type is OTHER to indicate when the standing order is performed.
TIME_3	(date). User-specified timestamp 3; used when the FREQUENCY type is OTHER to indicate when the standing order is performed.
TIME_4	(date). User-specified timestamp 4; used when the FREQUENCY type is OTHER to indicate when the standing order is performed.
TIME_5	(date). User-specified timestamp 5; used when the FREQUENCY type is OTHER to indicate when the standing order is performed.
TIME_6	(date). User-specified timestamp 6; used when the FREQUENCY type is OTHER to indicate when the standing order is performed.
RECORD_EXPIRED	(varchar 16). Date and time when this record becomes invalid.
RECORD_UPDATE	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
HOLIDAY_OFFSET	(char 1). Determines whether payment is generated before or after a non-business day. A = After or B = Before
CREATOR	(varchar 10). The operator who added the standing order.
UPDATER	(varchar 10). The operator who updated the standing order.
VERIFIER	(varchar 10). The operator who verified the standing order change.
NUM_MONTHS	(number 5). The Frequency for monthly or end-of-month payments.

Standing Order Log Table (STANDING_ORD_LOG_T)

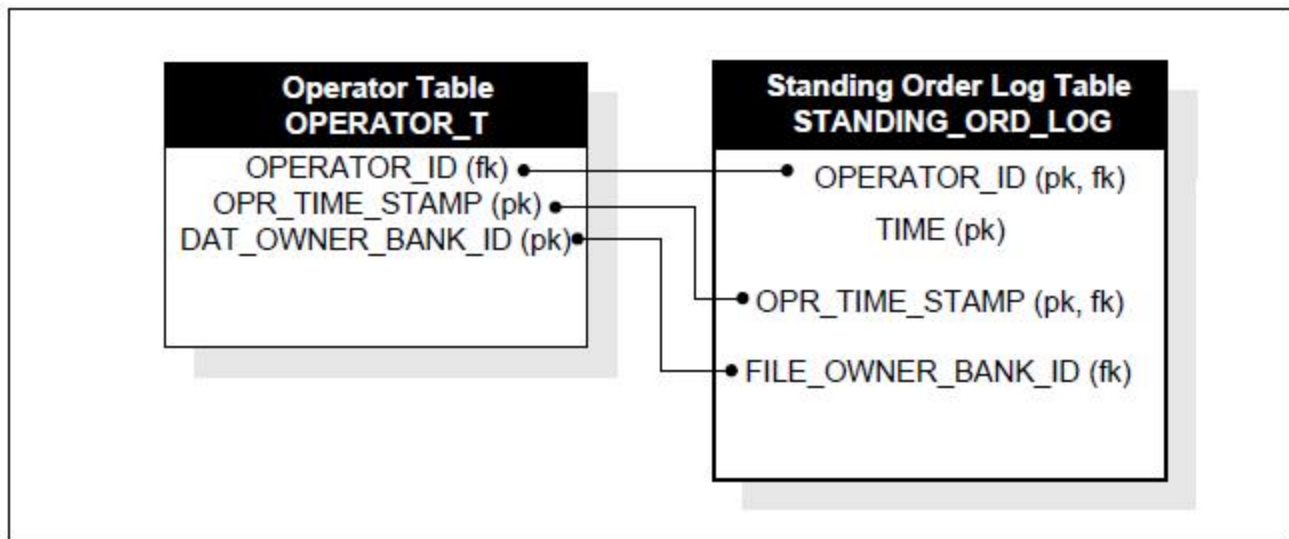
The Standing Order Log table provides data for the STO Activity report.

Table Relationships

The OPERATOR_ID and OPR_TIME_STAMP columns in the Standing Order Log table relate to the same columns in the Operator table.

Relationship Diagram

This relationship diagram shows the Standing Order Log table along with the Operator table, related to it through foreign keys.



Column Descriptions

OPERATOR_ID (pk, fk)	(varchar 12). Operator who performed the action.
TIME (pk)	(date). Date when the record was last updated.
SEQUENCE_NUMBER (pk)	(number 10). Unique identifier of the transaction.
OPR_TIME_STAMP (pk, fk)	(varchar 16). Date and time when the record was updated

Standing Order Tables

RECORD_EXPIRED	(varchar 16). Indicates whether this row's timestamp column has been propagated to previous rows.
RECORD_UPDATED	(varchar 1). Ensures that a different operator verifies the work of this operator.
PERSON	(varchar 10). ID of the operator who performed a function.
AMOUNT	(number 21.3). Amount involved in a transaction.
TEXT	(varchar 20). Message text received or sent with a message.
MEMO	(varchar 80). Descriptive text describing the operator action.
FILE_KEY_PROD_ID	(varchar 3). Product used by the operator who performed the action.
FILE_KEY_BANK_ID	(varchar 3). Bank ID of the bank where the operator uses MTS.
FILE_KEY_LOC	(varchar 6). Bank-defined location within the bank where the operator uses MTS.
FILE_KEY_CUST	(varchar 35). Customer ID.
FILE_KEY_NAME	(varchar 33). Customer name.
FILE_TYPE_PROD_ID	(varchar 3). Product used by the operator who performed the action.
FILE_TYPE_BANK_ID (fk)	(varchar 3). Bank ID of the operator who performed the action.
FILE_TYPE_LOC	(varchar 6). Location of the operator who performed the action.
FILE_TYPE_CUST	(varchar 20). Customer ID.
FILE_TYPE_NAME	(varchar 33). Name field from the primary index.
FILE_OWNER_PROD_ID	(varchar 3). Product used by the operator who performed the action.
FILE_OWNER_BANK_ID (fk)	(varchar 3). Bank ID of the operator who performed the action.
FILE_OWNER_LOC	(varchar 6). Location of the operator who performed the action.
FILE_OWNER_CUST	(varchar 20). Customer ID.
DAT_FUNC	(varchar 7). Function that the operator performed.

Static Operator Log Table (STATIC_OPR_LOG_T)

The Static Operator Log table records and stores information about any action operators have taken on static data.

Actions include adds, updates, verifies, and deletes. Static data, for example addresses, account information, and PINs, is not message related, but rather is data used to process the message.

Primary Keys

The Static Operator Log table (STATIC_OPR_LOG_T) has the following primary keys:

- QUEUE_LINE_ID
- OPERATOR_ID
- OPR_LOG_TIME_STAMP

Column Descriptions

QUEUE_LINE_ID (pk)	(varchar 12). Queue name that triggered the log event.
OPERATOR_ID (pk)	(varchar 10). The operator identifier that triggered the log event.
OPR_LOG_TIME_STAMP (pk)	(decimal 23). The date and time at which this log event was triggered.
RECORD_KEY	(varchar 65). Key of record that triggered the log event.
FUNCTION_NAME	(varchar 3). Name of the function performed when the log event was triggered.
FUNCTION_ACTION	(varchar 7). Name of the action performed when the log event was triggered.

Stop Tables

The Stop tables contain information that MTS uses to compare transactions on your system against the Office of Foreign Assets Control (OFAC) lists.

If a transaction contains a phrase in an OFAC list, MTS stops the transaction from processing any further. The Stop tables list the reason and category of a stop, the CHIPS UID or S.W.I.F.T. ID (if applicable), and a source number and time stamp for the stop.

Parent/child relationships

The Stop table (STOP_T) is the parent table to the following child tables:

- Stop Name (STOP_NAME_T)
- Stop Token (STOP_TOKEN_T)

Primary keys

The Stop tables have the following primary keys in common:

- STOP_SRC
- STOP_NUM
- STP_TIME_STAMP

Stop Table (STOP_T)

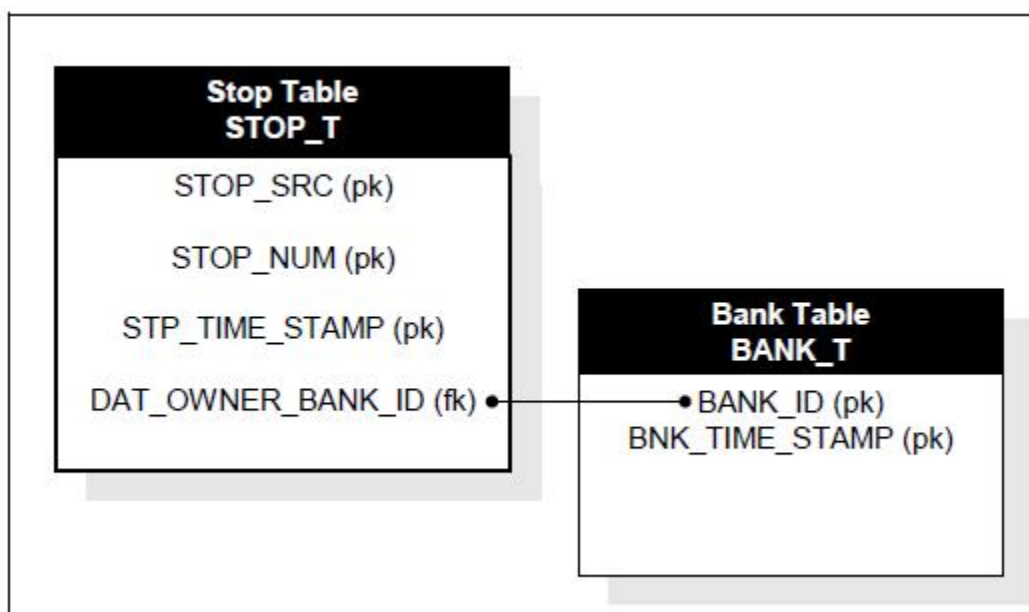
The Stop table provides summary information about the matching criteria for determining whether to stop a transaction.

Table Relationships

The DAT_OWNER_BANK_ID column in the Stop table relates to the BANK_ID column in the Bank table.

Relationship Diagram

This relationship diagram shows the Stop table along with the Bank table, related to it through foreign keys.



Column Descriptions

STOP_SRC (pk)	(varchar 3). Identifies the source of the stop or false hit entity; mapped from the SOURCE field on the STOP Display and Entry Screen. This column contains the following values: ENT (Entered manually) CAT (Obtained from a master verification record) CHP (Loaded from a CHIPS database)
---------------	---

Stop Tables

	JMA (Loaded from a JMA database)
STOP_NUM (pk)	(varchar 7). ID number MTS assigns a stop item when it is created; mapped from the REF # field on the STOP Display and Entry Screen.
STP_TIME_STAMP (pk)	(varchar 16). Date and time when the stop item was updated.
DAT_FUNCTION	(varchar 7). Function that last accessed the item in the Stop file; contains the following values: VFY ADD (Verify add) VFY DEL (Verify delete) VFY UPD (Verify update)
CREATOR	(varchar 10). Operator ID and bank ID of the operator who added the item to the Stop file.
CREATE_DATE	(date). Date when the item was added to the Stop file.
UPDATER	(varchar 10). Operator ID and bank ID of the operator who updated the item in the Stop file.
UPDATE_DATE	(date). Date when the item was updated in the Stop file.
VERIFIER	(varchar 10). Operator ID and bank ID of the operator who verified the item in the Stop file.
VERIFY_DATE	(date). Date when the item was verified in the Stop file.
DAT_OWNER_PROD_ID	(varchar 3). Product ID.
DAT_OWNER_BANK_ID (fk)	(varchar 3). Bank ID.
DAT_OWNER_LOC	(varchar 6). Location ID.
DAT_OWNER_CUST	(varchar 20). Customer ID.
CATEGORY	(varchar 30). Group (often a country). to which a stop item belongs.
REASON	(varchar 4). Identifies the reason for stopping transactions containing this stop entity. OFAC and BANK are currently the only possible reasons.
ENABLE_FLAG	(varchar 1). Flag indicating whether MTS uses a stop item's phrases during STOP processing; mapped from the ENABLED field on the STOP Display and Entry Screen. This column contains the following values: Y (MTS matches transactions against the phrases in this stop item during STOP processing. MTS sends transactions with text matching a phrase in a stop item to OFAC review.) N (MTS omits this item during STOP processing. MTS allows transactions with text matching phrases in these items to process normally.)
STOP_TYPE	(varchar 22). Type of phrase used as the stop phrase; contains the following values:

	NOTHING_IS (Stop value is not set) STOP_DESCRIPTOR_IS (Stop value is the descriptor) OVERRIDE_DESCRIPTOR_IS (Stop value is the override descriptor) EQUIVALENT_TOKEN_IS (Stop value is the equivalent token)
ORIGIN	(varchar 3). Origin of the stop item; contains the following values: ENT (Entered manually) CAT (Obtained from master verification record) CHP (Loaded from a CHIPS database) JMA (Loaded from a JMA database)
ENTITY_TYPE	(varchar 13). Type of entity. An entity is the name and address of an SDN (specially-designated national). . This column contains the following values: COUNTRY_IS (Entity is a country) INDIVIDUAL_IS (Entity is an individual) NOTHING_IS (Entity type is not set)
CHIPS_UID	(varchar 6). CHIPS universal ID of recipient.
SWIFT_ID	(varchar 30). S.W.I.F.T. ID.
RECORD_EXPIRED	(varchar 16). Date and time when this stop item becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.
TOKEN_SUMMARY	(varchar 78). Summary of stop tokens, for the STOP_INDEX. The values exported from STOP_INDEX.
MSG_DIRECTION	(varchar 8). Provides the ability to consider STOP hits for an entity based on message direction.

Stop Name Table (STOP_NAME_T)

The Stop Name table provides the list of stop names for a stop entity.

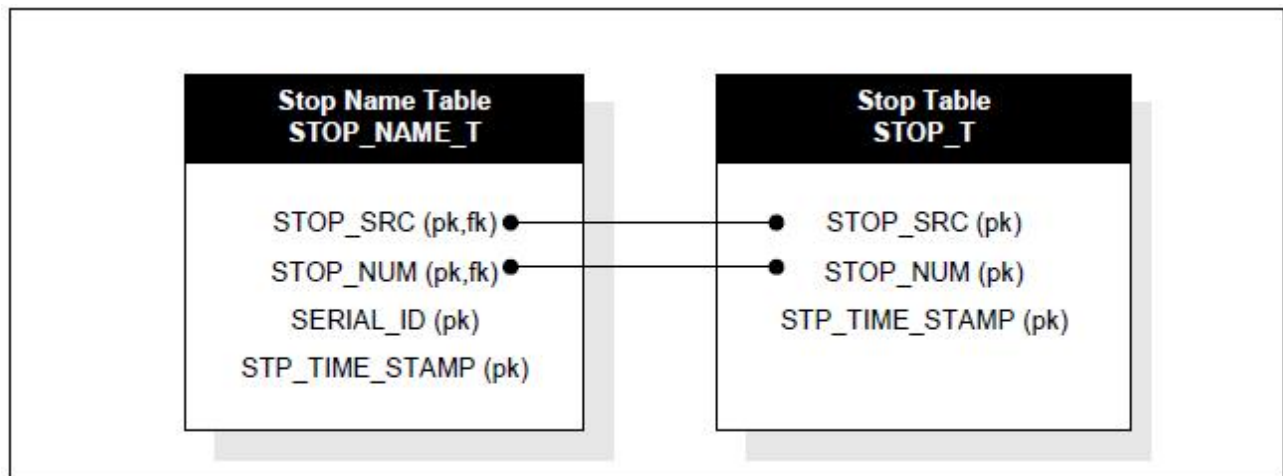
Table Relationships

The following columns in the Stop table relate to the same columns in the Stop Name table:

- STOP_SRC
- STOP_NUM

Relationship Diagram

This relationship diagram shows the Stop Name table along with the Stop table, related to it through foreign keys.



Column Descriptions

STOP_SRC (pk, fk)	(varchar 3). Identifies the source of the stop or false hit entity. This column contains the following values: ENT (Entered manually) CAT (Obtained from master verification record) CHP (Loaded from a CHIPS database) JMA (Loaded from a JMA database)
-------------------	--

STOP_NUM (pk, fk)	(varchar 7). Transaction number that MTS assigns a stop item when it is created.
SERIAL_ID (pk)	(number 19). Numeric ID that uniquely identifies this Stop Name in the STOP database. The serial ID ensures each row in the Stop Name table is unique. It is a unique numeric value assigned by Entia to sequence elements. The serial ID is ascending but not necessarily sequential.
STP_TIME_STAMP (pk)	(varchar 16). Date and time when the stop item was updated.
STOP_NAME	(varchar 120). User-defined stop phrase; mapped from the PHRASES field on the STOP DATA screen.
RECORD_EXPIRED	(varchar 16). Date and time when this stop item becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Stop Token Table (STOP_TOKEN_T)

The Stop Token table provides the list of stop tokens associated with each stop phrase.

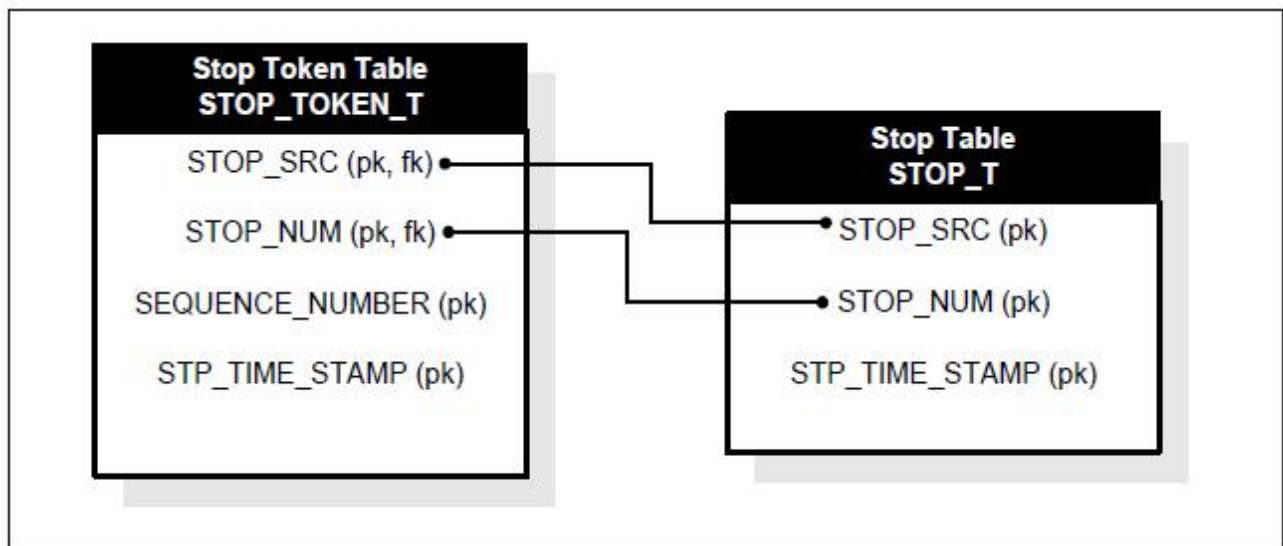
Table Relationships

The following columns in the Stop table relate to the same columns in the Stop Token table:

- STOP_SRC
- STOP_NUM

Relationship Diagram

This relationship diagram shows the Stop Token table along with the Stop table, related to it through foreign keys.



Column Descriptions

STOP_SRC (pk, fk)	(varchar 3). Identifies the source of the stop or false hit entity. This column contains the following values: ENT (Entered manually) CAT (Obtained from master verification record) CHP (Loaded from a CHIPS database)
-------------------	--

	JMA (Loaded from a JMA database)
STOP_NUM (pk, fk)	(varchar 7). Transaction number that MTS assigns a stop item when it is created.
SEQUENCE_NUMBER (pk)	(number 10). Unique number that identifies a token in the STOP database.
STP_TIME_STAMP (pk)	(varchar 16). Date and time when the stop item was updated.
TOKEN_ID	(varchar 60). Individual word in a stop entity. MTS generates a token for each word entered in the ENTITY field on the STOP DATA screen.
INCLUDE_FLAG	(varchar 1). Flag indicating whether to include the stop token (X). or not (blank) during stop processing.
RECORD_EXPIRED	(varchar 16). Date and time when this stop item becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Supplement ID Table (SUPPLEMENT_ID_T)

The Supplement ID table contains additional IDs that can be attached to a message. Supplement IDs (formerly known as Extended IDs) are used by some clearing systems to identify accounts that don't fit the standard account structure.

Primary Keys

The Supplement ID table (SUPPLEMENT_ID_T) has the following primary keys:

- FUNC_AREA
- FUNC_AREA_ID_TYPE
- FUNC_AREA_ID
- IDBANK
- IDTYPE
- IDACC
- IDADR
- IDPAD
- SUPP_TIME_STAMP

Column Descriptions

FUNC_AREA (pk)	(varchar 3). Functional area of the database.
FUNC_AREA_ID_TYPE (pk)	(char 1). ID type within the functional area of the database.
FUNC_AREA_ID (pk)	(varchar 30). ID within the functional area of the database.
IDBANK (pk)	(varchar 3). ID bank.

Supplement ID Table (SUPPLEMENT_ID_T)

IDTYPE (pk)	(char 1). ID type for account indices.
IDACC (pk)	(varchar 30). Account number for the account type address indices.
IDADR (pk)	(varchar 5). Address code for the account type address indices.
IDPAD (pk)	(char 1). Padding for the account type address indices.
SUPP_TIME_STAMP (pk)	(number 16). Date and time when the record was updated.
EXT_ID_IDTYPE	(char 1). Extended ID type, for example 'S' (SWIFT BIC). .
EXT_ID_SUBTYPE	(varchar 2). Extended ID subtype, for example BL as in E/BL.
EXT_ID_TAG	(varchar 6). Extended ID tag used for matching an address associated ID.
MEMO	(varchar 80). Descriptive text describing the operator action.
CREATE_SOURCE	(varchar 80). Creation source; source of the supplement ID.
RECORD_EXPIRED	(number 16). Date and time when this stop item becomes invalid.
RECORD_UPDATED	(char 1). Indicates whether this row's timestamp column has been propagated to previous rows.

S.W.I.F.T. Authentication Table (SWIFT_AUTH_T)

The S.W.I.F.T. Authentication table provides information about the keys your bank uses to authenticate messages exchanged with S.W.I.F.T. correspondent banks. For each correspondent bank, you can specify active (current) and future (next) bi-lateral keys and their effective dates.

Primary keys

The S.W.I.F.T. Authorization table (SWIFT_AUTH_T). has the following primary keys:

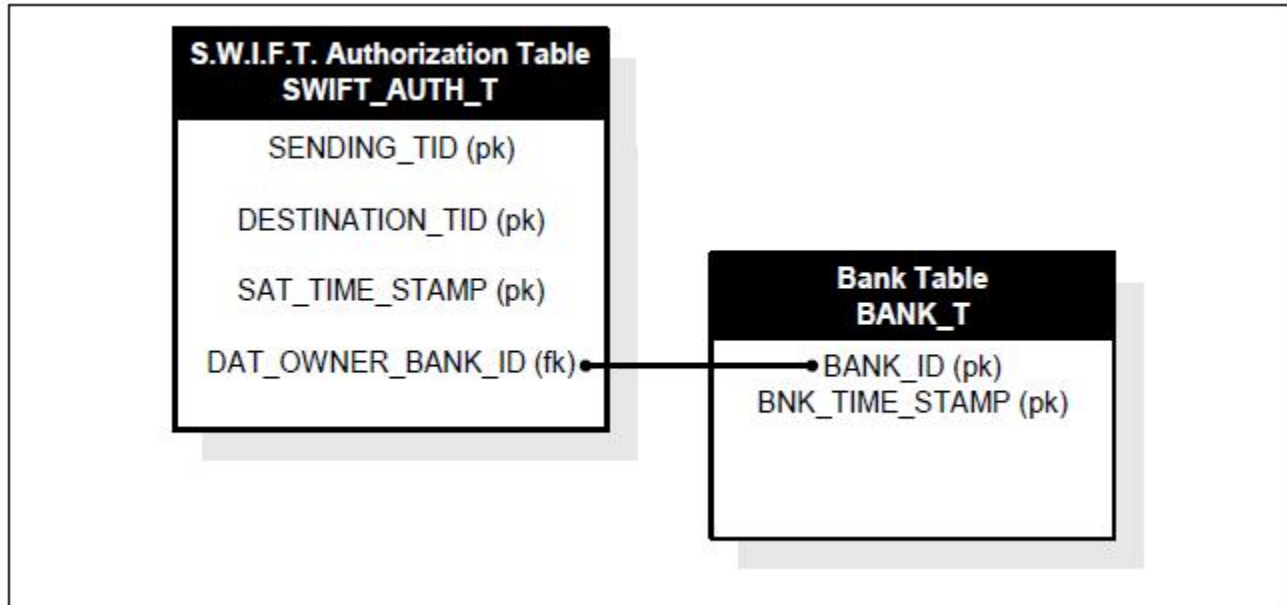
- SENDING_TID
- DESTINATION_TID
- SAT_TIME_STAMP

Table Relationships

The DAT_OWNER_BANK_ID column in the S.W.I.F.T. Authentication table relates to the BANK_ID column in the Bank table.

Relationship Diagram

This relationship diagram shows the S.W.I.F.T. Authentication table along with the Bank and Operator tables.



Column Descriptions

SENDING_TID (pk)	(varchar 11). Sending TID (terminal identifier) for this authenticator.
DESTINATION_TID (pk)	(varchar 11). S.W.I.F.T. destination TID (terminal identifier) for this authenticator.
SAT_TIME_STAMP (pk)	(varchar 16). Date and time when the item was updated.
CREATOR	(varchar 10). Operator ID and bank ID of the operator who created the record.
CREATE_DATE	(date). Date when the record was created.
UPDATER	(varchar 10). Operator ID and bank ID of the operator who last updated the record.
UPDATE_DATE	(date). Date when the item was updated.
VERIFIER	(varchar 10). Operator ID and bank ID of the operator who verified the record.
VERIFY_DATE	(date). Date when the item was verified.
DAT_OWNER_PROD_ID	(varchar 3). Product ID.
DAT_OWNER_BANK_ID (fk)	(varchar 3). Bank ID.

S.W.I.F.T. Authentication Table (SWIFT_AUTH_T)

DAT_OWNER_LOC	(varchar 6). Location ID.
DAT_OWNER_CUST	(varchar 20). Customer ID.
ADR1	(varchar 35). First line of the address.
ADR2	(varchar 35). Second line of the address.
ADR3	(varchar 35). Third line of the address.
ADR4	(varchar 35). Fourth line of address.
INFORMATION	(varchar 20). Informational column.
IN_KEY_ORIG	(varchar 4). Inbound bilateral key originator; contains the following values: US or THEM.
OUT_KEY_ORIG	(varchar 4). Outbound bilateral key originator; contains the following values: US or THEM.
PREAGREEMENT	<p>(varchar 11). Status of the bilateral key relationship described by this S.W.I.F.T. Authentication (SAT). record. This column contains the following values:</p> <p>CONFIRMED (BKE pre agreement data has been acknowledged and okayed by the remote bank.)</p> <p>DELETED (The bilateral key relationship has been deleted and needs to be deleted from the Authenticator File in the branches.)</p> <p>EXCLUDED (The destination is not allowed to exchange either BKE messages or authenticated traffic. If the authentication process finds a relationship with this correspondent status, authentication automatically fails.)</p> <p>SUSPENDED (The exchange of authenticated traffic with a correspondent has been stopped temporarily, even though keys are present and can continue to be exchanged. If the authentication process finds a relationship with this correspondent status and the end-date is equal to zero, or before or equal to today, authentication automatically fails.)</p> <p>UNCONFIRMED (BKE pre agreement data has not been confirmed by the bank.)</p>
EXCHANGE_TYPE	<p>(varchar 11). Type of key exchange; contains the following values:</p> <p>AUTOMATED (Key exchange is performed electronically by sending an MT960).</p> <p>MANUAL (Key exchange is performed using paper, mail, or phone).</p>
KMA_USR	(varchar 11). Fully-qualified key management authority (KMA) address for the local destination.
KMA_COR	(varchar 11). Fully-qualified key management authority (KMA) address for the remote or correspondent destination.
BK_TYPE_USR	<p>(varchar 1). Local bilateral key type; required if the EXCHANGE_TYPE is AUTOMATED. This column contains the following values:</p> <p>0 (Bidirectional, controlled by lower address)</p> <p>1 (Bidirectional, controlled by higher address)</p>

S.W.I.F.T. Authentication Table (SWIFT_AUTH_T)

	2 (Unidirectional, controlled by lower address) 3 (Unidirectional, controlled by higher address)
BK_TYPE_COR	(varchar 1). Remote or correspondent bilateral key type; required if the EXCHANGE_TYPE is AUTOMATED. This column contains the following values: 0 (Bidirectional, controlled by lower address) 1 (Bidirectional, controlled by higher address) 2 (Unidirectional, controlled by lower address) 3 (Unidirectional, controlled by higher address)
RENEW_FREQ_USR	(number 5). Key renewal frequencies, in months (locally-controlled key).
RENEW_FREQ_COR	(number 5). Key renewal frequencies, in months (remotely-controlled key).
BKE_TIME_USR	(date). Last successful bilateral key exchange (BKE). timestamp, (locally-initiated exchange).
BKE_TIME_COR	(date). Last successful bilateral key exchange (BKE). timestamp, (remotely-initiated exchange).
CTP_USR	(number 19). Locally-controlled exchange counter; increments each time a key is exchanged.
CTP_COR	(number 19). Remotely-controlled exchange counter; increments each time a key is exchanged.
BK_ID_OLD_OUT_PFX	(varchar 1). Prefix of the official S.W.I.F.T. format key identifier for the previous outgoing bilateral key. This value is always B.
BK_ID_OLD_OUT_TYPE	(varchar 1). Key type of the previous outgoing bilateral key; contains the following values: 0 (Bidirectional, controlled by lower address) 1 (Bidirectional, controlled by higher address) 2 (Unidirectional, controlled by lower address) 3 (Unidirectional, controlled by higher address)
BK_ID_OLD_OUT_DATE	(varchar 6). Effective date of the previous outgoing bilateral key.
BK_ID_OLD_OUT_KCV	(varchar 8). System-calculated key check value for the previous outgoing bilateral key.
BK_ID_CUR_OUT_PFX	(varchar 1). Prefix of the official S.W.I.F.T. format key identifier for the current outgoing bilateral key. This value is always B.
BK_ID_CUR_OUT_TYPE	(varchar 1). Key type of the current outgoing bilateral key; contains the following values: 0 (Bidirectional, controlled by lower address) 1 (Bidirectional, controlled by higher address) 2 (Unidirectional, controlled by lower address)

S.W.I.F.T. Authentication Table (SWIFT_AUTH_T)

	3 (Unidirectional, controlled by higher address)
BK_ID_CUR_OUT_DATE	(varchar 6). Effective date of the current outgoing bilateral key.
BK_ID_CUR_OUT_KCV	(varchar 8). System-calculated key check value for the current outgoing bilateral key.
BK_ID_NEW_OUT_PFX	(varchar 1). Prefix of the official S.W.I.F.T. format key identifier for the next outgoing bilateral key. This value is always B.
BK_ID_NEW_OUT_TYPE	(varchar 1). Key type of the next outgoing bilateral key; contains the following values: 0 (Bidirectional, controlled by lower address) 1 (Bidirectional, controlled by higher address) 2 (Unidirectional, controlled by lower address) 3 (Unidirectional, controlled by higher address)
BK_ID_NEW_OUT_DATE	(varchar 6). Effective date of the next outgoing bilateral key.
BK_ID_NEW_OUT_KCV	(varchar 8). System-calculated key check value for the next outgoing bilateral key.
BK_ID_OLD_IN_PFX	(varchar 1). Prefix of the official S.W.I.F.T. format key identifier for the previous incoming bilateral key. This value is always B.
BK_ID_OLD_IN_TYPE	(varchar 1). Key type of the previous incoming bilateral key; contains the following values: 0 (Bidirectional, controlled by lower address) 1 (Bidirectional, controlled by higher address) 2 (Unidirectional, controlled by lower address) 3 (Unidirectional, controlled by higher address)
BK_ID_OLD_IN_DATE	(varchar 6). Effective date of the previous incoming bilateral key.
BK_ID_OLD_IN_KCV	(varchar 8). System-calculated key check value for the previous incoming bilateral key.
BK_ID_CUR_IN_PFX	(varchar 1). Prefix of the official S.W.I.F.T. format key identifier for the current incoming bilateral key. This value is always B.
BK_ID_CUR_IN_TYPE	(varchar 1). Key type of the current incoming bilateral key; contains the following values: 0 (Bidirectional, controlled by lower address) 1 (Bidirectional, controlled by higher address) 2 (Unidirectional, controlled by lower address) 3 (Unidirectional, controlled by higher address)
BK_ID_CUR_IN_DATE	(varchar 6). Effective date of the current incoming bilateral key.
BK_ID_CUR_IN_KCV	(varchar 8). System-calculated key check value of the current incoming bilateral key.

S.W.I.F.T. Authentication Table (SWIFT_AUTH_T)

BK_ID_NEW_IN_PFX	(varchar 1). Prefix of the official S.W.I.F.T. format key identifier for the next incoming bilateral key. This value is always B.
BK_ID_NEW_IN_TYPE	(varchar 1). Next incoming bilateral key type; contains the following values: 0 (Bidirectional, controlled by lower address) 1 (Bidirectional, controlled by higher address) 2 (Unidirectional, controlled by lower address) 3 (Unidirectional, controlled by higher address)
BK_ID_NEW_IN_DATE	(varchar 6). Effective date of the next incoming bilateral key.
BK_ID_NEW_IN_KCV	(varchar 8). System-calculated key check value for the next incoming bilateral key.
NEW_IN_DATE	(date). Date and time when the next incoming bilateral key becomes effective.
NEW_OUT_DATE	(date). Date and time when the next outgoing bilateral key becomes effective.
CUR_IN_DATE	(date). Date and time when the current incoming bilateral key becomes effective.
CUR_OUT_DATE	(date). Date and time when the current outgoing bilateral key becomes effective.
OLD_IN_DATE	(date). Date and time when the previous incoming bilateral key became effective.
OLD_OUT_DATE	(date). Date and time when the previous outgoing bilateral key became effective.
NEW_IN_DEL	(varchar 1). New In Delete flag.
CUR_IN_DEL	(varchar 1). Currently In Delete flag.
OLD_IN_DEL	(varchar 1). Old In Delete flag.
NEW_OUT_DEL	(varchar 1). New Out Delete flag.
CUR_OUT_DEL	(varchar 1). Currently Out Delete flag.
OLD_OUT_DEL	(varchar 1). Old Out Delete flag.
SND_START_DATE	(date). Start date for the manually-entered send key pending verification.
SND_END_DATE	(date). End date for the manually-entered send key pending verification.
RCV_START_DATE	(date). Start date for the manually-entered receive key pending verification.
RCV_END_DATE	(date). End date for the manually-entered receive key pending verification.
RECORD_EXPIRED	(varchar 16). Date and time when this S.W.I.F.T. authentication record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Template Tables

Template tables contain information about templates that either work with individual operator profiles or in place of them.

Banks may choose their set up to include both or either. Templates contain sets of operator privileges and can be attached to multiple operator IDs in a single operation. Templates provide a means of giving multiple operators the same privilege set that can then be easily updated and maintained.

Primary keys

All template tables have the following primary keys in common:

- TEMPLATE_ID
- TPL_BANK
- TPL_LOC
- TPL_TIME_STAMP

Template Table (TEMPLATE_T)

The Template table contains information for the storage location of a particular template.

Primary Keys

The Template table (TEMPLATE_T) has the following primary keys:

- TEMPLATE_ID
- TPL_BANK
- TPL_LOC
- TPL_TIME_STAMP

Column Descriptions

TEMPLATE_ID (pk)	(varchar 10). Template ID
TPL_BANK (PK)	(varchar 3). Bank ID.
TPL_LOC (pk)	(varchar 6). Template location.
TPL_TIME_STAMP (pk)	(number 16). Date and time when the template record was updated.
DAT_FUNCTION	(varchar 7). Function that last accessed this template record; contains the following values: VFY ADD (Verify add) VFY DEL (Verify delete) VFY UPD (Verify update)

CREATOR	(varchar 10). Operator who created the template record.
CREATE_DATE	(date). Date on which the template record was created.
UPDATER	(varchar 10). Operator who updated the template record.
UPDATE_DATE	(date). Date on which the template record was updated.
VERIFIER	(varchar 10). Operator who verified the template record.
VERIFY_DATE	(date). Date on which the template record was verified.

Template Tables

DAT_OWNER_PROD_ID	(varchar 3). Product ID.
DAT_OWNER_BANK_ID	(varchar 3). Bank ID.
DAT_OWNER_LOC	(varchar 6). Location ID.
DAT_OWNER_CUST	(varchar 20). Customer ID.
ENABLED_FLAG	(varchar 1). A Y/N flag that indicates whether the template is enabled.
TPL_NAME	(varchar 25). Name of the template.
TPL_OPR_USE_COUNT	(number 5). Number of operators currently using this template.
RECORD_EXPIRED	(number 16). Date and time when this record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Template Privilege Table (TEMPLATE_PRIV_T)

The Template Privilege table contains information on the operator privileges that point to a specific template with which an operator may be associated.

Primary keys

The Template Privilege table (TEMPLATE_PRIV_T) has the following primary keys:

- TEMPLATE_ID
- TPL_BANK
- TPL_LOC
- PRODUCT_ID
- PRIVILEGE
- TPL_TIME_STAMP

Column Descriptions

TEMPLATE_ID (pk)	(varchar 12). Template ID.
TPL_BANK (pk)	(varchar 3). Bank associated with the template.
TPL_LOC (pk)	(varchar 6). Template location.
PRODUCT_ID (pk)	(varchar 3). Product ID.
PRIVILEGE (pk)	(varchar 8). Operator privilege.
TPL_TIME_STAMP (pk)	(number 16). Date and time when the record was updated.
SEQUENCE_NUM	(number 10). Number (counter). that uniquely identifies this entry.
RECORD_EXPIRED	(number 16). Date and time when this S.W.I.F.T. authentication record becomes invalid.
RECORD_UPDATED	(varchar 1). Indicates whether this row's timestamp column has been propagated to previous rows.

Index

A

Account Address Table [16](#)
Account Drawdown Table [21](#)
Account Due-From Balance Table [19](#)
Account Group Member Table [30](#)
Account Group Table [26](#)
Account Subaccount Table [23](#)
Account Table [9](#)
Account Tables [8](#)
ACCOUNT_ADDRESS_T [16](#)
ACCOUNT_DFM_BAL_T [19](#)
ACCOUNT_DRAWDOWN_T [21](#)
ACCOUNT_GRP_MBR_T [30](#)
ACCOUNT_GRP_T [26](#)
ACCOUNT_SUB_ACC_T [23](#)
ACCOUNT_T [9](#)
Address Destination Table [42](#)
Address PIN Privilege Table [49](#)
Address Table [32](#)
Address Tables [32](#)
Address Test History Table [50](#)
ADDRESS_DEST_T [42](#)
ADDRESS_PIN_PRIV_T [49](#)
ADDRESS_T [32](#)
ADDRESS_TST_HST_T [50](#)
ANT_TMPL_CR_T [180](#)
ANT_TMPL_DR_T [189](#)
ANT_TMPL_T [170](#)
Anticipation Template Debit Table [189](#)
Anticipation Template Table [170](#)
Anticipation Template Tables [169](#)
Anticipation Templates [169](#)
Anticipations [169](#), [170](#), [180](#), [189](#)
Anticipations Template Credit Table [180](#)
AUX [158](#), [167](#), [168](#)
AUX Tables [158](#)
AUX_ABA_DDA_T [167](#)
AUX_ADDRESS_T [159](#)
AUX_NOTES_T [168](#)
Auxiliary ABA DDA Table [167](#)
Auxiliary Address Table [159](#)
Auxiliary Notes Table [168](#)
Auxiliary Tables [158](#)

B

Bank Clearinghouse Table [203](#)
Bank Fee Table [209](#)
Bank Location Cost Table [208](#)
Bank Special Table [205](#)
Bank Table [199](#)
Bank Tables [198](#)
BANK_CLEARHOUSE_T [203](#)

BANK_LOC_COST_T [208](#)
BANK_SPECIAL_T [205](#)
BANK_T [199](#)
BNK [198](#), [199](#), [203](#), [205](#), [208](#), [209](#)

C

Change Data Table [213](#)
Change Exclusion Table [215](#)
Change List Table [212](#)
Change Tables [211](#)
CHANGE_DATA_T [213](#)
CHANGE_TBL_EXC_T [215](#)
Channel Address Type Table [220](#)
Channel Table [217](#)
Channel Tables [216](#)
CHANNEL_ADRTYP_T [220](#)
Channel_T [217](#)
CONFIG_ITEM_DATA_T [228](#)
CONFIG_ITEM_DESC_T [226](#)
CONFIG_ITEM_T [224](#)
CONFIG_T [222](#)
Configuration Item Data Table [228](#)
Configuration Item Description Table [226](#)
Configuration Item Table [224](#)
Configuration Table [222](#)
Configuration Tables [221](#)
Currency [230](#)
Currency Table [230](#)
CURRENCY_T [230](#)
CUSIP [232](#)
CUSIP Table [232](#)
CUSIP_T [232](#)

D

DFM [233](#), [234](#), [236](#), [238](#)
DFM_BAL_T [236](#)
DFM_DERIVED_T [238](#)
DFM_T [234](#)
DFM_TRN_T [240](#)
Due From Monitor [233](#), [234](#), [236](#), [240](#)
Due From Monitor Balance Table [236](#)
Due From Monitor Derived Table [238](#)
Due From Monitor Table [234](#)
Due From Monitor Tables [233](#)
Due from Monitor Transactions Table [240](#)

E

End-of-Day [118](#)
Exchange Rate Table [242](#)
Exchange Rates [242](#)

EXCHANGE_RATE_T 242

F

FEE_T 209

File Change Data Loader 211, 212, 213

File change Reports 211, 212

Funds Transfer Message Tables 54

M

Mandate Table 321, 329

MANDATE_T 321, 329

Message Accounting Table 98

Message ACH Regulatory Reporting Table 320

Message Batch Control Messages Fund Table 325

Message BCM Table 327

Message Credit Table 73

Message Debit Table 86

Message End of Day Table 118

Message History Table 111

Message Processing Rules Parameter Table 123

Message Processing Rules Table 120

Message Processing Rules Tables 120

Message Related Amount Table 117

Message Table 54

Message Tables 53

Message Text Table 115

MESSAGE_ACCTG_T 98

MESSAGE_ACH_RRPT_T 320

MESSAGE_BCM_FUND_T 325

MESSAGE_BCM_T 327

MESSAGE_CR_T 73

MESSAGE_DR_T 86

MESSAGE_EOD_T 118

MESSAGE_HIST_T 111

MESSAGE_PR_PRM_T 123

MESSAGE_PR_T 120

MESSAGE_T 54

MESSAGE_TEXT_T 115

Monitor Table 247

Monitor_T 247

Monitoring 247

MTS Configuration 221, 222, 224, 226, 228

O

OFAC (Office of Foreign Assets Control) 338

Operator Change Table 256

Operator Error Table 259

Operator Log Table 261

Operator Logon Table 263

Operator Privilege Table 265

Operator Profile Amount Table 267

Operator Table 254

Operator Tables 253

Operator Template Table 269

OPERATOR_CHANGE_T 256

OPERATOR_ERROR_T 259

OPERATOR_LOG_T 261

OPERATOR_LOGON_T 263

OPERATOR_PRIV_T 265

OPERATOR_PROFAMT_T 267

OPERATOR_T 254

OPERATOR_TMPL_T 269

P

Period Date Table 271

PERIOD_DATE_T 271

PIN Account Table 277

PIN Function Override Table 279

PIN Function Table 281

PIN Table 273

PIN Tables 272

PIN_ACCOUNT_T 277

PIN_FUNC_OVR_T 279

PIN_FUNCTION_T 281

PIN_T 273

PINs 272

Processing Rule Table 284

Processing Rules 120, 123, 283, 284, 286, 288

Processing Rules Match Table 286

Processing Rules Parameter Table 288

Processing Rules Parameter Values Table 289

Processing Rules Tables 283

Profile Table 290

PROFILE_T 290

Profiles 290

PRULE_MATCH_T 286

PRULE_PARAM_T 288

PRULE_PARAM_VL_T 289

PRULE_T 284

PSR Table 292

PSR_T 292

R

Relational Gateway Link Table 301

Relationship File Tables 7

Remote Parameter Table 308

Remote Parameters 302

Remote Table 303

Remote Tables 302

REMOTE_PARAMETER_T 308

REMOTE_T 303

Repetitive Credit Table 134

REPETITIVE DEBIT TABLE 146

Repetitive Messages 125

Repetitive Table 125

Repetitive Tables 125

REPETITIVE_CR_T 134

REPETITIVE_DR_T 146

REPETITIVE_T 125

RGWLINK_T 301

Rule From Name Table 314

Rule Nickname Table 316

Rule Pattern Table 317

Rule Statistics Table 318

Rule Table 311

Index

RULE_FROM_NAME_T [314](#)
RULE_NICKNAME_T [316](#)
RULE_PATTERN_T [317](#)
RULE_STATS_T [318](#)
RULE_T [311](#)
Rules Tables [310](#)

S

S.W.I.F.T. Authentication Table [348](#)
SEPA [319](#), [321](#), [325](#), [329](#)
Standing Order Log Table [335](#)
Standing Order Table [331](#)
Standing Order Tables [330](#)
STANDING_ORD_LOG_T [335](#)
STANDING_ORDER_T [331](#)
Static Operator Log Table [337](#)

STATIC_OPR_LOG_T [337](#)
Stop Table [339](#)
Stop Tables [338](#)
Stop Token Table [344](#)
STOP_T [339](#)
STOP_TOKEN_T [344](#)
Supplement ID Table [346](#)
SUPPLEMENT_ID_T [346](#)
SWIFT_AUTH_T [348](#)

T

Template Privilege Table [357](#)
Template Table [355](#)
Template Tables [354](#)
TEMPLATE_PRIV_T [357](#)
TEMPLATE_T [355](#)